

IMPROVING INTELLIGENT ANALYTICS THROUGH GUIDANCE:
ANALYSIS AND REFINEMENT OF PATTERNS OF USE AND
RECOMMENDATION METHODS FOR DATA MINING AND ANALYTICS
SYSTEMS

by

JEREMY R. PATE

BRANDON DIXON, COMMITTEE CHAIR
TRAVIS ATKISON
DAVID BROWN
ALLEN PARRISH
RANDY SMITH

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2019

Copyright Jeremy R. Pate 2019
ALL RIGHTS RESERVED

ABSTRACT

In conjunction with the proliferation of data collection applications, systems that provide functionality to analyze and mine this resource also increase in count and complexity. As a part of this growth, understanding how users navigate these systems, and how that navigation influences the resulting extracted information and subsequent decisions becomes a critical component of their design. A central theme of improving the understanding of user behavior and tools for their support within these systems focuses the effort to gain a context-aware view of analytics system optimization. Through distinct, but interwoven, articles this research examines the specific characteristics of usage patterns of a specific example of these types of systems, construction of an educational support system for new and existing users, and a decision-tree supported workflow optimization recommender system. These components combine to yield a method for guided intelligent analytics that uses behavior, system knowledge, and workflow optimization to improve the user experience and promote efficiency of use for systems of this type.

LIST OF ABBREVIATIONS

ADVANCE	Advanced Dashboard for Visualization Analysis and Coordinated Enforcement
CART	Classification and Regression Tree
CARE	Critical Analysis Reporting Environment
CSV	Comma Separated Values
GUID	Globally Unique Identifier
EDM	Educational Data Mining
HE	High Engagement
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
LA	Learning Analytics
LT	Linger Time
MCC	Matthews Correlation Coefficient
OTR	Operation Take Rate
PRNG	Pseudorandom Number Generator
RI	Recommendation Impact
RQ	Research Question
SPA	Single Page Application

TEL	Technology Enhanced Learning
TOUR	Tree Optimized User Recommendations
TSP	Time Spent on Page
WUM	Web Usage Mining
YAML	Yet Another Markup Language

ACKNOWLEDGEMENTS

I would like to thank my parents, Jerry and Wanda, for their guidance and my wife, Jessica, for her patience and diligent proofreading.

I would also like to thank my doctoral committee, Dr. Brandon Dixon, Dr. David Brown, Dr. Allen Parrish, Dr. Travis Atkison, and Dr. Randy Smith for their feedback and support throughout this process.

CONTENTS

ABSTRACT	ii
LIST OF ABBREVIATIONS	iii
ACKNOWLEDGEMENTS	v
LIST OF TABLES	ix
LIST OF FIGURES	x
CHAPTER 1 INTRODUCTION	1
CHAPTER 2 ARTICLE 1 - ADVANCE-ING ANALYTICS: INVESTIGATION OF PATTERNS OF USER IN A PUB- LIC SAFETY ANALYTICS SYSTEM	5
2.1 Introduction	5
2.2 Research Background and Questions	6
2.3 Data Collection	7
2.4 Analysis	9
2.4.1 RQ1: Session Characteristics	10
2.4.2 RQ2: Session Engagement	14
2.4.3 RQ3: Age and Duration	21

2.5	Conclusions and Future Work	27
	References	30
CHAPTER 3	ARTICLE 2 - FAST (RE)INTRODUCTION TO ANALYTICS DATASETS: NOVEL AND POPULIST NAVIGATION USING USER SOURCED PATHS	32
3.1	Introduction	32
3.2	Related Research	33
3.3	Data Collection and Path Construction	34
3.4	Recommendation Methodology	38
3.4.1	Novel Navigation	39
3.4.2	Populist Navigation	41
3.5	Recommendation Setup and Experiment	43
3.6	Results	46
3.6.1	Candidate and Presented Recommendations	48
3.6.2	Categories and Take Rates	51
3.7	Conclusions and Future Work	55
3.7.1	Novel Method and <i>Already Recommended</i>	55
3.7.2	Potential Practical Benefits	57
3.7.3	Future Work	57
	References	59
CHAPTER 4	ARTICLE 3 - TOUR GUIDE: PROVIDING WORKFLOW OPTIMIZED RECOMMEN- DATIONS TO USERS OF AN ANALYTICS SYSTEM	61

4.1	Introduction	61
4.2	Research Background	62
4.2.1	System Usage Extraction	63
4.2.2	System Usage Analysis and Motivation	66
4.3	TOUR Guide	69
4.3.1	Feature Extraction	70
4.3.2	Decision Tree Construction and Export	72
4.3.3	Recommendation Generation	76
4.4	Recommendation Simulation Results	78
4.5	Conclusions and Future Work	86
4.5.1	Conclusions	86
4.5.2	Future Work	88
	References	90
CHAPTER 5 CONCLUSION		92
5.1	Understanding Overall User Behavior	92
5.2	Education and Awareness	94
5.3	Preference	95
5.4	Recommendations and Optimization	97
5.5	Considerations and Summary	99
REFERENCES		101

LIST OF TABLES

2.1	ADVANCE options considered for <i>operations</i> . Modification of any of these values marks them for inclusion in the set.	8
2.2	Calculated Age and Duration values	25
3.1	User operation and context database table - <i>tblOperationLog</i>	36
3.2	User operation parameters table - <i>tblOperationLogParameters</i>	37
3.3	Operation categories, as extracted to the Markov chain.	46
3.4	Markov chain construction results.	46
3.5	Combined recommendation given/taken disparity.	52
4.1	Description of extracted features.	71

LIST OF FIGURES

2.1	Screenshot of <i>ADVANCE</i> . The <i>dataset filters</i> are displayed in the left column and the <i>dataset display preferences</i> are displayed in the four large panels to the right.	9
2.2	Diagram of the logging process for <i>ADVANCE</i>	9
2.3	Length of the user's session, measured in seconds, shown as a percentage occurrence of all sessions.	12
2.4	Operations per session with <50 operations, shown as percentage of total sessions with that count. This graph represents 94.12% of the total sessions, with sessions above 50 operations making up the remainder. Sessions above 50 operations are not shown and are not included in the percentage calculation (for clarity).	13
2.5	Chart displaying length of sessions by number of operations per session. . . .	15
2.6	Operation linger time for 0 to 10 seconds of rounded linger time.	15
2.7	Operation linger time for 1 to 10 of rounded linger time (note that operations with less than 0.5 seconds of linger time (marked as zero in Figure 2.6) have been removed).	16
2.8	Plot of location in session with linger time of operation. Note the regular distribution of linger times throughout the portions of the session.	20
2.9	Chart of high engagement operations, listed by placement in the total session length.	21
2.10	Chart of measured difference in low and high engagement. Positive percentages indicate that high engagement occurred at a higher comparative rate than low engagement at that point.	21
2.11	Chart of high engagement operations for sessions with 1-10 operations, listed by placement in the total session length.	22

2.12	Chart of high engagement operations for sessions with 11-25 operations, listed by placement in the total session length.	23
2.13	Chart of high engagement operations for sessions with 26+ operations, listed by placement in the total session length.	24
2.14	Graph of Age and Duration. Larger circles represent a larger occurrence of that Age/Duration intersection.	25
2.15	K-Means Cluster Analysis of Age and Duration - 3 clusters.	27
2.16	K-Means Cluster Analysis of Age and Duration - 4 clusters.	28
3.1	Screenshot of <i>ADVANCE</i> . The <i>dataset filters</i> are displayed in the left column and the <i>dataset display preferences</i> are displayed in the four large panels to the right.	33
3.2	Diagram of the logging process for <i>ADVANCE</i>	35
3.3	Portion of the Markov chain describing the state transitions between variables.	38
3.4	Recommendation construction process for Novel Navigation.	40
3.5	Recommendation construction process for Populist Navigation.	42
3.6	Combined - Recommendation results by action category (<i>Already Done</i> , <i>Already Recommended</i> , and <i>Recommendation Given</i>), shown as a percentage of the total calculated recommendations.	47
3.7	Breakdown of each portal's total volume contribution to the total recommendations given as seen in the <i>combined</i> recommendations.	48
3.8	Populist recommendation candidate results.	49
3.9	Novel recommendation candidate results.	50
3.10	Take rates of Novel and Populist methods for each portal.	52
3.11	Combined - Novel - Given and Taken recommendation categories (<i>Chart</i> , <i>Data Source</i> , <i>Date</i> , <i>Filter</i> , <i>Hide Null Values</i> , <i>Other</i> , and <i>Variable</i>), listed as percentages of the total Given and Taken recommendations, respectively.	53

3.12	Combined - Populist - Distribution of Given and Taken recommendation categories (<i>Chart, Data Source, Date, Filter, Hide Null Values, Other, and Variable</i>), listed as percentages of the total Given and Taken recommendations, respectively.	53
3.13	Combined take rate distribution by category and Novel and Populist methods.	54
3.14	Novel categorical take rates, listed by portal.	56
3.15	Populist categorical take rates, listed by portal.	56
4.1	Diagram of the logging process for ADVANCE.	65
4.2	Screenshot of <i>ADVANCE</i> . The <i>dataset filters</i> are displayed in the left column and the <i>dataset display preferences</i> are displayed in the four large panels to the right.	66
4.3	Session export position for all sessions (single and multiple).	68
4.4	Session export position for sessions with multiple exports.	68
4.5	Session export position for sessions with one export only.	69
4.6	Subset of a full tree (with export specificity capped at 7 levels).	73
4.7	Summary of the Feature Evaluation Process.	79
4.8	Distribution of the decision tree recommendations by category.	80
4.9	Distribution of the Recommendation Impact (RI) frequency in the total recommendation set.	80
4.10	Distribution of the recommendations taken, by placement in the overall session.	83
4.11	Distribution of the raw in-session take rates.	83
4.12	Distribution of the normalized in-session impact-adjusted take rates.	84

CHAPTER 1

INTRODUCTION

As the adoption of robust digital data collection systems increases, the volume and complexity of the datasets that are the results of those efforts represent a vast potential source of information. However, even with advanced analysis systems, the complex and nuanced questions that users use these systems to answer are often difficult to resolve without significant knowledge of the data itself. Each dataset presents a challenging but alluring potential goldmine, and the tools with which to find the most valuable vein are as important as the tools to extract it.

Tools and methods to allow users *of all experience levels* to be effective data driven decision makers is vital. Tools like the Advanced Dashboard for Visualization Analysis and Coordinated Enforcement (ADVANCE)[6] system provide a set of data exploration and examination utilities for large and complex datasets, primarily for government and public safety users. These users make wide use of these data, from answering direct data questions to hotspot analysis to determine optimal officer patrol routes[7]. Within this group, the assumption or requirement that experienced and trained users will be given the time to attain familiarity with these advanced datasets is not guaranteed. The quickening pace at which data are being used as the critical component of a decision that has far-reaching impacts can promote an environment of results-driven decisions. As the demand for more decisions rises, the supply of experts in those datasets must also rise. Unfortunately, expertise takes time to acquire. A method and system to support *fast* and *accurate* data

analysis can allow for relatively inexperienced users to be guided through the data without the requirement that this training take place over months and years.

Using the research and related tools in the area of recommender systems[3] to determine and provide ideal recommendations for a set of user-focused goals, we can provide users a tool to digitally augment their navigation of the system. The overall goal is to develop a system which will guide the user to an actionable decision based on the data. The system will use the data available from the collected actions, inherent domain and statistical information, and any other available information to support this process. From this goal, a number of interrelated research questions emerge:

- How can user intent be determined?
- What aspects of the dataset are useful in determining relevant recommendations?
- How can user preference for an item be determined?
- Does an explicitly determined intent improve the recommender system's efficiency at making correct recommendations?

With these questions in mind, a method and subsequent implemented system to *determine the user's goal and maximize the system's support of that goal through relevant and accurate recommendations* is the primary focus of this research. Once the method and system are established and tested, any additional data that are collected (through normal use of the system) will feed the improvement of the model.

A secondary and practical goal of this research is to find the most effective method(s) of providing recommendations to users of the ADVANCE-type analytics systems that maximize the decision making effectiveness (based on the user's goals) and minimizes time wasted. It is important to recognize that no two users are the same, and that many worthwhile avenues can be pursued during the exploration of data. For example, for some users, the decision making process is most supported by a seemingly meandering navigation of

the data. While this meandering may seem unguided and random at first, the user may be asking and answering multiple questions while using the system with little demarcation between the activities for each question. For others, a single, aptly recommended page is the best solution. Determining the intents of these two very different users and providing relevant recommendations to their goals is a multifaceted challenge, with several smaller problems to solve in that pursuit.

The components of this work can be broken into three primary areas of research within this system and this domain:

1. **Understand** - Comprehending user behavior examination and classification
2. **Educate** - Educating users through enhanced/augmented learning
3. **Optimize** - Providing workflow identification and optimization

For this first area, the goal is to understand how users navigate through the system. This behavior takes several forms, but a major focus is to identify and characterize the page navigation patterns to extract a set of usable and accurate user behavior paradigms, from which potential session optimization methods could be applied.

For the second area, we look to the problem of education and training within the dataset itself. While this also involves the system in which the analytics activities are taking place, the major focus is on the user's familiarity with the corpus of knowledge presented by the datasets within the system, including both popular and unpopular pieces of that data.

For the third, and final, area we want to determine patterns of workflows within the system and optimize the system to provide recommendations that meet that workflow. This is especially focused on those that either directly indicate or at least strongly suggest user preference. Beginning by identifying what emergent workflows exist and demonstrating at least some indication of preference, then examining a path-related method by which to direct users toward items along that optimal workflow path, we construct a complete workflow that works to nudge the user in an identified optimal direction.

These components all share a core diagnostic dataset for the ADVANCE data analysis system[6], which has not seen examination of this type prior to these studies. User session behavior data was collected by examining analytics system usage by logging calls to a backend data service which handles the analytics processing. These activity logs demonstrate the user and system (automated) activity generated by the frontend system. These data are then used to create a sequence-aware graph of user action over the course of their session. We use implicit feedback due to the relatively small number of users (approximately 2400).

Overall, this work seeks to understand multiple aspects of the use of an analytics data system, particularly in the area of public safety data and by public sector users. Within this domain, the topics of investigation were related to overall user behavior, educational support, and workflow optimization. As such, this work is organized into three primary articles, each of which identify and investigate facets of data mining and analytics systems. The first of these articles deals primarily with an in-depth examination of user behavior, specifically within their distinct visits to the site (i.e. sessions). The second article examines a recommender-system approach to dataset user education through use of a Markov chain. The final article explains a novel, path-based approach for providing user recommendations for items that were shown to be preferred by users on a path that ended in an export of data.

Each article is presented as a titled and self-contained work of research, with internally cited references, figures, equations, and tables. As such, each article is intended to be interpreted somewhat independently. However, the overall context for the articles is within the goal of providing a comprehensive (but not exhaustive) examination of user behavior and potential optimization methods for that behavior, within the scope of a public safety analytics system.

CHAPTER 2

ARTICLE 1 - ADVANCE-ING ANALYTICS: INVESTIGATION OF PATTERNS OF USER IN A PUBLIC SAFETY ANALYTICS SYSTEM

2.1 Introduction

ADVANCE[15] (Advanced Dashboard for Visualization Analysis and Coordinated Enforcement) is a web-based analytics portal used for examination of relatively large (tens of millions of records) datasets, primarily through criteria-based set manipulation and reduction to select down to the desired filtered data elements. This system is based on the CARE analytics engine [12] that uses a custom process for extracting, translating, and loading data into a format that allows for extremely fast analysis of relationally stored datasets, which are common in the public safety domain. This system is used by myriad types of users, including law enforcement officers, epidemiologists, roadway engineers, and other state government personnel. Data analyzed and displayed by the system includes ambulance run reports, vehicle crash reports, and citations.

One of the primary use cases for this system is to allow for approachable analysis and exploration of complex data by a relative layperson in statistics and data analytics. This process, rooted in a knowledge discovery[3] background, provides a tool to extract the data necessary to build towards knowledge, and ultimately, informed conclusions based on that knowledge. More specifically, this system is used for data-based decision making. These

can include direct data related questions or more complex scenarios, such as using historical crash data and hotspot analysis for officer patrol routing[17].

As the volume and complexity of collected data increases, tools for examining and distilling the lessons presented by this data are extremely important. Intuition and “gut feelings”, even those based on years of experience in a field dealing with the activities that create these datasets, are potentially deceptive and can be inaccurate or misleading. These can exist for many reasons, include common misconceptions, ‘data myths’, or biases that exist within an organization.

Using diagnostic usage data collected for telemetry and debug purposes, we have web service call summary data for analytics session within the system. Through examination of this data, we discovered several overall patterns, some of which are counter to previously held assumptions about system use.

2.2 Research Background and Questions

Web usage monitoring ([1], [6], and [16]), utilizes several techniques for extraction and analysis of web usage data for the purpose of determining usage patterns. Along with the act of simply extracting the data (which is often not so simple), the processing of the resulting web usage logs to even extract a workable dataset is an arduous task.

The use of *time spent on web pages* (TSP) as a measure of interest or preference is well documented within the are of web usage mining (WUM)[8, 10]. Given that ADVANCE is a single-page application (SPA), page linger time measured as a change in the state of actually served pages is not an accurate description of the user’s activity. Because of this, we use the slightly modified concept of linger time (LT) as a viewing preference metric.

In-depth analysis of specialty analytics systems are often performed internal to an organization, and may or may not distribute any lessons learned to the wider research community. Part of our efforts stem from conceptualizing the users of these types of systems not only as professional analysts, but also as students of the data. This approach con-

textualizes the users as learners within this domain, and has roots in learning analytics (LA)[14] and educational data mining (EDM)[2]. In this way, we also want to learn how to best establish an analytics system as a trusted ‘teacher’[5] in that it both reveals and drives the user toward a set of conclusions, either intentionally or unintentionally.

In an effort to understand the nature of how this system is used, and how that use compares to other systems, we established the following overall research questions. These questions are primarily focused on two factors: session linger analysis and dataset parameters for time-quanta analysis.

RQ1: Is there a relationship between session length and session operation count? (i.e. do longer sessions occur as the result of simply more operations, or something else?)

RQ2: How can we define periods of high user engagement, and where do those periods occur during the session?

RQ3: Are there any patterns in the dataset analysis time parameters that might imply a preference or possible motivation to modify the default?

In order to expand the knowledge on how these systems are utilized, and to seek out possible methods that would improve and promote an adaptable system[4], we began by collecting and curating a reasonably complex dataset with which we could begin our investigation.

2.3 Data Collection

In the ADVANCE system (and its variants), users enter a secure login portal to access the data analytics system. After successfully logging in, they are presented with a screen that looks like the one shown in Figure 2.1. For reference, the primary modifiable variables which the user can modify to affect the main analysis screen are listed in Table 2.1.

There are two primary categories of this collected data: *dataset filters* and *dataset display preferences*. Dataset filters refers to parameters such as start and end date, predefined

Table 2.1: ADVANCE options considered for *operations*. Modification of any of these values marks them for inclusion in the set.

Option	Description
<i>Start Date</i>	Date to begin analysis.
<i>End Date</i>	Date to end analysis.
<i>Filter</i>	Pre-created (often multi-variable) set reduction parameter (e.g. construction workzone involved vehicle crashes)
<i>Filter Variable</i>	Parameter to further limit data (e.g. county or state agency)
<i>Chart 1-4 Variable</i>	Displayed variable in the chosen chart window (of the 4 available).
<i>Chart 1-4 Type</i>	Displayed chart type (e.g. bar, pie, or table)

filters (built manually by the creators of the datasets), and one or more filtering criteria (chosen from the available categorical variables of the dataset). These are shown in the left pane of Figure 2.1. The dataset display preferences, shown in the four large panels of Figure 2.1, are composed of the four available variable display tiles and the different display methods (graph types) that can be used to visualize that data. Once the user has used the filters to pare down the dataset, they can then explore the frequencies of up to four unique variables in that set.

This diagnostic data consists of approximately 14,000,000 web service calls made by the ADVANCE frontend to the CARE web service analytics backend. However, only roughly 5% of the calls (around 750,000) are actual user-initiated actions. The remainder are either system-initiated calls or related to a real-time location mapping service, which generates a very large amount of automated requests whose results are used to update a displayed map. These calls are logged as shown in Figure 2.2, with the operation and parameters being captured in separate, but correlated database tables.

In order to remove the ambiguity that is inherent with time, IP address, or user-based session delimiters[7, 9], the diagnostic data also includes an explicit session identifier, reducing the effects of cross-session metrics on the final due to improper session break identification. This session identifier is collected as a globally-unique identifier (GUID), and is assigned by the logging subsystem.

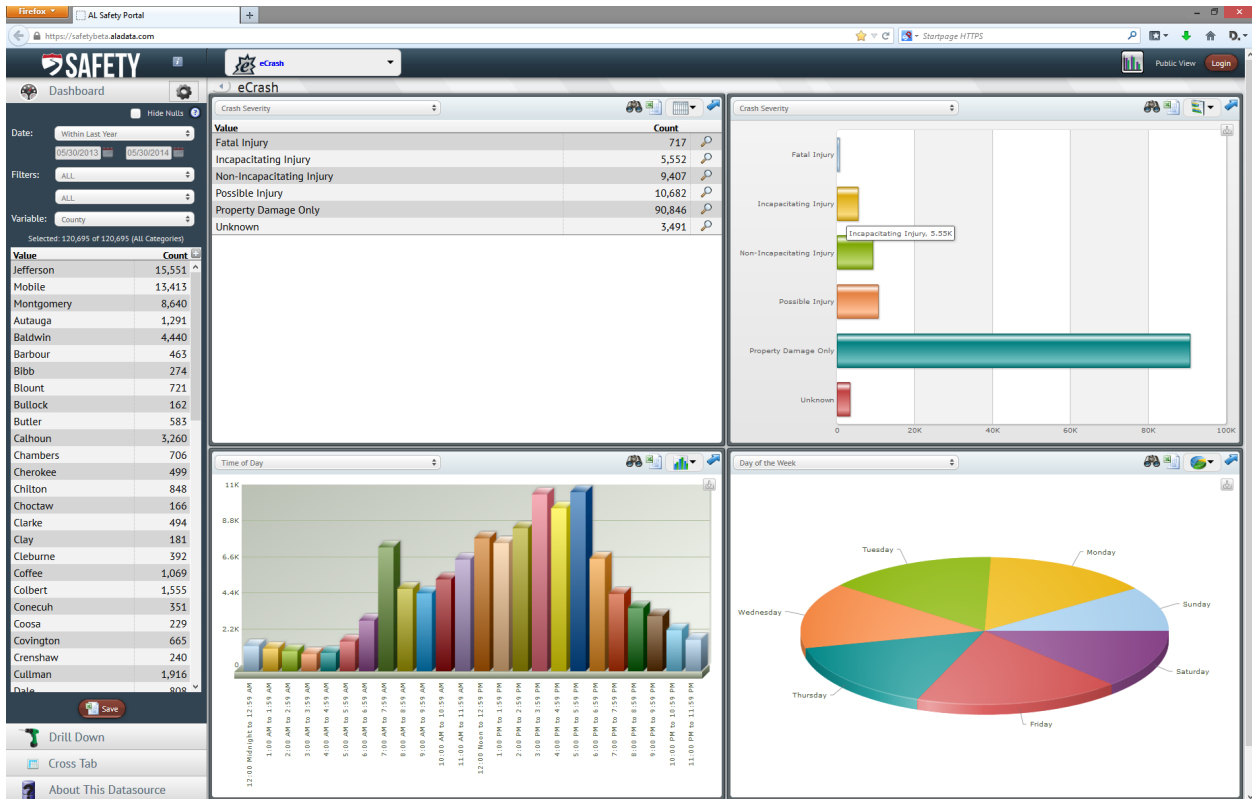


Figure 2.1: Screenshot of *ADVANCE*. The *dataset filters* are displayed in the left column and the *dataset display preferences* are displayed in the four large panels to the right.

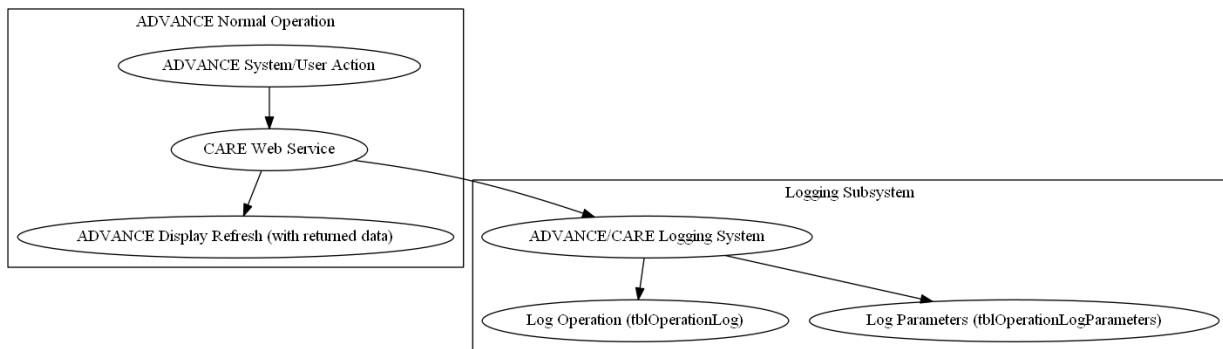


Figure 2.2: Diagram of the logging process for *ADVANCE*.

2.4 Analysis

In order to answer the posed research questions, an analysis of the use of the system was performed, with a primary focus on any time-based implicit preference indicators that were available. For this analysis, we did not consider the content related to the choices

during each user’s session, only the choices related to general loiter time in the system. However, we do intend to (in a later experiment) explore patterns and potential recommendations related to the types of values. This choice was made primarily to expand the general value of the analysis, as the specific choice of variables may not be generally useful outside of their domain. To aid in clarity for the remainder of the paper, below are a few term definitions which will be used to describe the summary data:

- **Session** - The contiguous period in which the user is active in the system. A 30-minute (1,800 second) authentication timeout was in place for all ADVANCE portals, meaning that any no-activity period lasting longer than this timeout was reset as a separate session, as the user would be required to re-login.
- **Operation** - Any analytics user-initiated analytics action.
- **Linger Time (LT)** - Time between operations during the session. This is measured by the timestamps of the web service calls, which are initiated by mouse clicks in the interface. This metric is also used as a component of the calculation of engagement.

2.4.1 RQ1: Session Characteristics

Our first research question: “Is there a relationship between session length and session operation count?”, seeks to address whether the relationship between the user’s time spent in the system is related to operations, linger time, or possibly another factor. To answer the question as to whether a relationship exists between session operation length and session count, we begin by examining the average user’s experience in the system. Over all of the candidate sessions, the average session consists of approximately **14 user analytics operations** taking place over an average of **216 seconds (3.6 minutes)**, with an average of **15.4 seconds between each operation**. This average session, though not necessarily a typical one, does show that sessions are relatively brief, but active (with around 4

operations each minute). However, through further analysis, the deviations from this average demonstrate some interesting patterns of use.

Overall Session Length Users of the ADVANCE system have a significant amount of data exploration options, spread across multiple-sourced datasets. Given the amount of options, examination of the amount of time the user spends during each contiguous visit (i.e. their *session*) provides information about not only the typical user behavior, but also provides a baseline for comparison in determining any session-position related behavior patterns. For this analysis, the total session length was calculated by summing all of the linger times for the operations. This calculation is expressed in Equation 2.1, where i is the 1-based index of all operation linger times in a session.

As seen in Figure 2.3, roughly 70% of the sessions are under 90 seconds. The distribution of the remaining 30% sessions is very spread out over the total number of sessions. This long tail of high session lengths results in a standard deviation of session length of 341.32 seconds, and (as stated before) an average session length of 216 seconds.

Based on this data, it appears that the majority of users prefer briefer sessions, though a small (but not insubstantial) subset spends much longer in the system, even up to the point of a typical session timeout. There were points in the data that showed extreme values for session length (some as high as 14 hours), but these were discarded, as they were either the result of the constantly updated map display keeping the session active, or some other authentication malfunction (several sessions of this type had no operations for over the time that the authentication timeout would have taken effect).

$$Session\ Length = \sum_{i=1}^n Linger\ Time\ (LT)_i \quad (2.1)$$

Operations Per Session Next, we looked at the number of operations that occurred during a user's session. Note that sessions that contained less than 2 operations were not

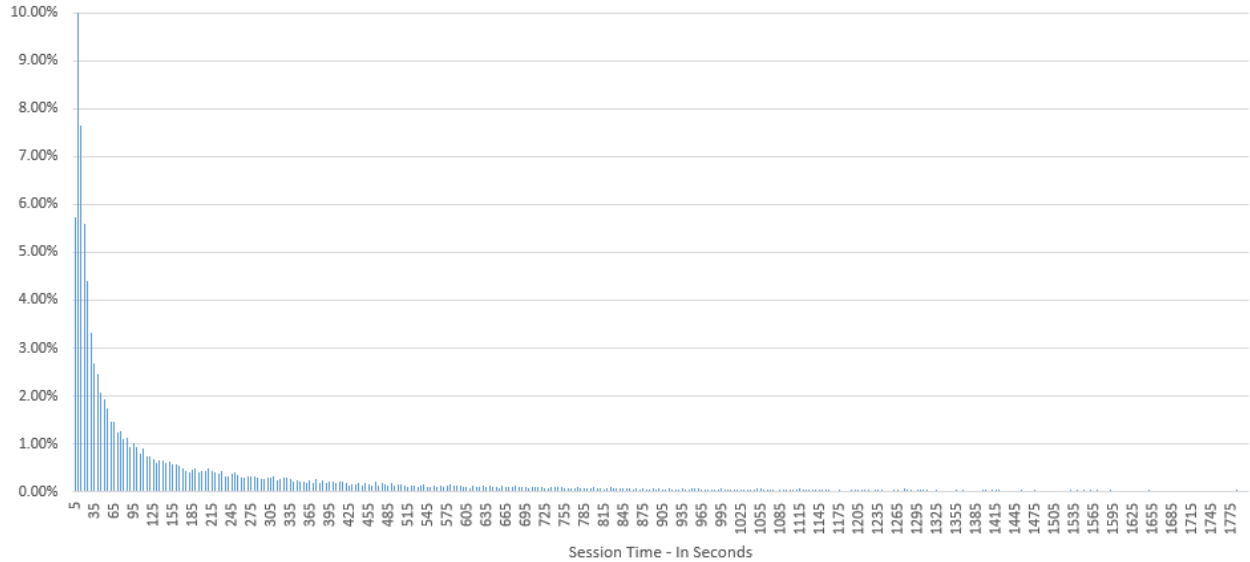


Figure 2.3: Length of the user’s session, measured in seconds, shown as a percentage occurrence of all sessions.

considered, as the majority of the operations with a single session had linger times (which will be discussed later) that indicated a user had logged in and immediately terminated the session (possibly due to a network issue).

The results, shown in Figure 2.4 show a clear pattern of early, clustered operations with a sharp drop after roughly 10 operations in the session. The overwhelming majority of like users are clustered this way, with approximately 70% of the sessions consisting of 10 or less operations. Interestingly, there is a significant spike at 6 (and especially 7) operations in a session, with this occurrence representing 30% of all sessions. We theorize that this is possibly representative of the basic “single round” analysis, wherein a user has selected a start date (1), end date (1), filter (1), filter variable (1), and at least one each of the chart types and variables (2-3), for a sum of 6 or 7 (depending on the specific choices made). This also suggests that users engage with the system in a way that may be consistent with the design, and possibly indicates that most users trend toward a targeted analysis of a predetermined set of parameters. Regardless, further analysis of the relationship between this occurrence and the specific variable breakdown is likely warranted.

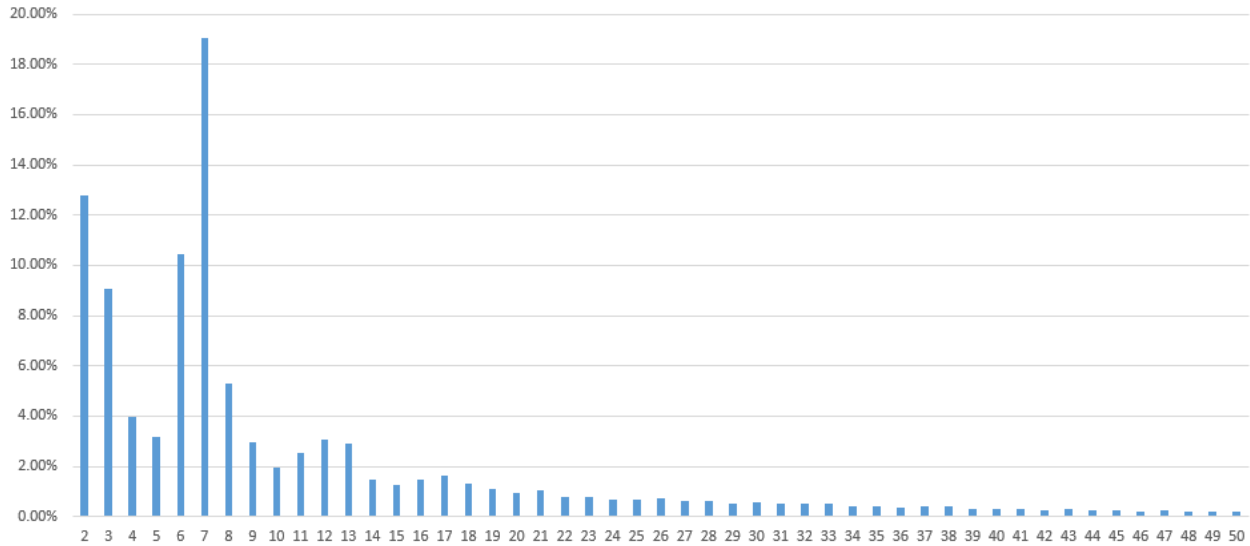


Figure 2.4: Operations per session with <50 operations, shown as percentage of total sessions with that count. This graph represents 94.12% of the total sessions, with sessions above 50 operations making up the remainder. Sessions above 50 operations are not shown and are not included in the percentage calculation (for clarity).

Operations and Session Length Another interesting relationship to investigate is that of any potential correlation between operations and total session length, and whether patterns of operation linger time changes over sessions of differing lengths. This relationship is also important to understand because of the potentially associated relationship of session length to average linger time.

In theory, the more operations that the user performs (and as the amount of data seen by the user increases), the longer the user will need to evaluate the results of those actions and continue to make more navigation choices in the system. This pattern can be seen as an extension of the necessity of the requirement of more time (i.e. reading a full article before being able to determine whether a particular item is actually relevant)[10]. To determine whether there was an immediately discernible relationship, the Pearson correlation coefficient is used as a simple litmus test for the relationship of many of these factors (calculated as shown in Equation 2.2). At first glance, there does appear to be a relationship between number of operations and total session time, albeit a weak one. Based on the

value, 0.51921114 we can only claim a moderate positive relationship between the number of operations and the length of the session.

$$C = \frac{n \sum xy - \sum x \sum y}{\sqrt{(n \sum x^2 - (\sum x)^2)(n \sum y^2 - (\sum y)^2)}} \quad (2.2)$$

Upon further inspection, a significant break in behavior was found at approximately the 115 operation mark. Above this mark, with only 204 sessions, the standard deviation for the session length was 385 seconds (and an average of 1146 seconds). Below this mark, with a 39,368 sessions, the standard deviation is 334 seconds (and an average of 211 seconds). Based on this 13.3% difference that came from only 0.05% of the sessions, the data were re-examined, excluding sessions with operations greater than 115 operations. Because of a consistent level of deviation between sessions in this grouping, the average session length was calculated for each distinct value of operation count. This was especially important given that there were over 400,000 operations within this set, which made visualization difficult outside of an average grouping. This exclusion of operations greater than 115 operations yielded much more consistent results, as shown in Figure 2.5, with a resulting calculated determination coefficient (R^2) of 0.9167 for a calculated trendline for the average session length. This demonstrates a solid linear correlation between the number of operations and session length. Conversely, this also indicates that linger time is not affected by the length of the session. Overall, we can find no evidence that sessions are longer other than because of more operations performed by the user.

2.4.2 RQ2: Session Engagement

With our second research question: “How can we define periods of high user engagement, and where do those periods occur during the session?” we hope to differentiate typical TSP from LT, as well to further identify and examine the characteristics of especially examined parts of the system through the use of a new metric.

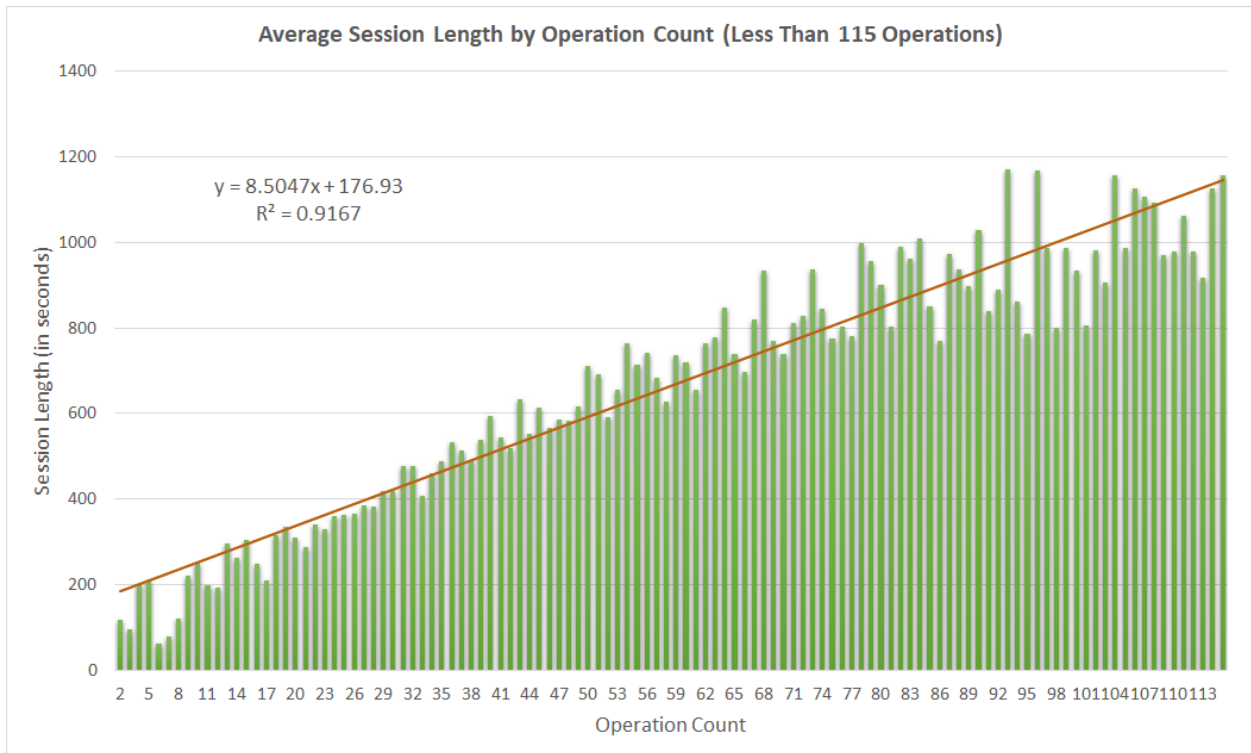


Figure 2.5: Chart displaying length of sessions by number of operations per session.

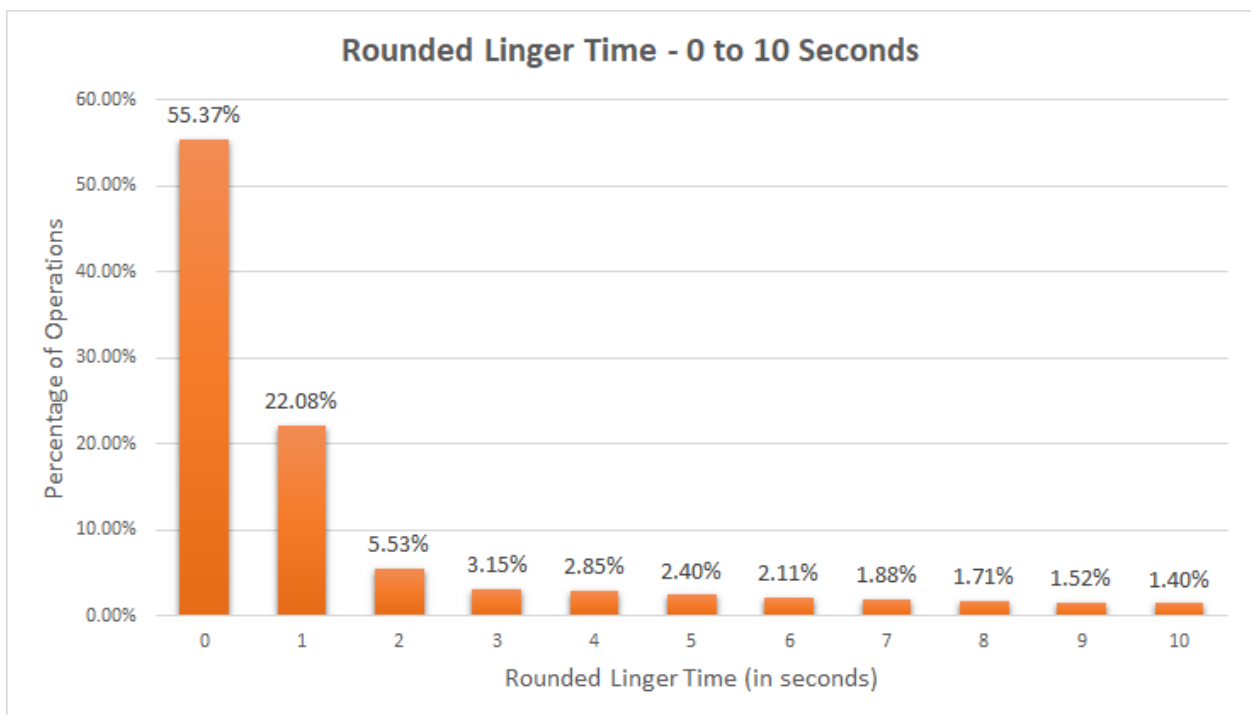


Figure 2.6: Operation linger time for 0 to 10 seconds of rounded linger time.

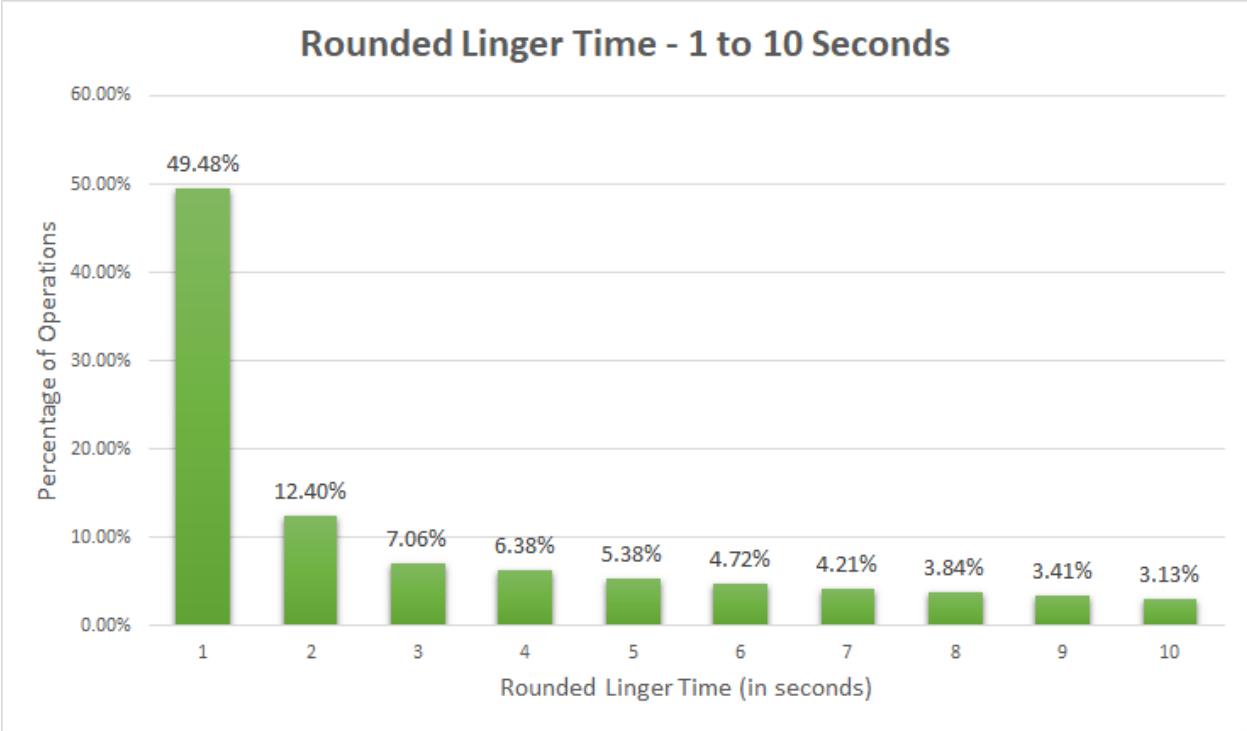


Figure 2.7: Operation linger time for 1 to 10 of rounded linger time (note that operations with less than 0.5 seconds of linger time (marked as zero in Figure 2.6) have been removed).

Before examining linger time (and subsequently engagement) over the entire session, we first explored how linger time presented over the population of single operations. The analysis is divided into two sub-results: with 0 (<0.5 seconds) and without 0 (starting at 1 second). This division was made due to the overwhelming occurrence of sub 0.5 second linger times (55% of all operations). Linger times extend well beyond 10 seconds, with times all the way up to the session timeout cutoff (1,800 seconds). However, beyond the 10 second mark, the distribution pattern is very similar to that shown for the session (2.1), with most occurrence percentages in the sub 1% range.

Considering that over 70% of the operations are from 0 to 1.49 seconds (marked as 0 and 1), and that operations longer than this make up the remainder of the set (30%), it appears that most of the operations are spent in configuration to yield the desired result, which is then examined over a wide range of times (beyond the 2 second mark). This is supported, especially for the sub-0.5 second mark, due to the result return time, which can

be in the 0.05 to 0.1 second range (depending on the dataset size and total server load at the time). With this in mind, it becomes important to determine if there are any points in the population of user sessions that represent reoccurring and consistent instances of *high engagement*.

To determine if there are any patterns of relative increase in user interest (*high engagement*) in a point in the session, the linger time for operations were extracted throughout the session and plotted by their position in the total time spent in that session. This process normalizes the occurrence for both long and short sessions, and avoids operation index related overemphasis. For example, an operation with index 5 would be marked at 20% in the session for a session of operation length 10 and 10% in a session of operation length 20. The results of this linger time plot can be seen in Figure 2.8. For simplicity (and to avoid chatter in the data) these linger times were broken into 0.05 (5%) increments of the session by rounding the operation placement value. Note that values listed as 0 on this chart represent values below 2.5% in the session. As previously stated, several behaviors were observed at this point in the session, but this value was allowed to remain in the plot since it is being used primarily as a visual reference and not a statistical one.

The visual distribution of linger time over the course of the session does not show any significant breaks, suggesting that linger time varies in a regular fashion throughout the session for the user population. There are intermittent breaks, especially starting above the 800 second session mark, but no gaps of over 100 seconds can be seen except at the 0%, 5%, and 25% mark. Also, the correlation coefficient for the relationship between the session location and the linger time is 0.064091917 . This suggests that linger time is not linearly correlated to placement in the session, and that longer sessions do not necessarily result in higher linger times.

Following this initial discovery, we sought to identify a basic metric to identify an operation as *high engagement*. For this, we examined each session independently, calculating the mean and standard deviation of the linger time of the operations for each session. We

considered sessions as individual instances to avoid variations in individual user behavior (e.g. a tendency to look at items for very brief or very long periods). Considering each session independently, we mark each operation as *high engagement* if it satisfies the criteria defined in Equation 2.3. Additionally, the data were considered as percentages *relative to the instance of each category (low and high engagement) independently at each point, with the total percentage of each category calculated within that category only*. This was done to avoid having the extremely high rate of low engagement operations overshadow any high engagement values to the point of becoming invisible when placed on the chart. These values show the *relative frequency* of the occurrence of low and high engagement operations, but are not directly related to total frequency over all operations at that point in the session.

The results of this calculation over the set of all operations is shown in Figure 2.9 as a combined graph of the placement in the session of both the high and low engagement linger events. Of all recorded operations, only 5.24% were calculated as being *high engagement*. Within this data, only four session locations had instances where the percentage occurrence of high engagement was greater than that of low/normal engagement (shown in Figure 2.10) 0.3, 0.35, 0.6, and 1. Of these, only two occurrences (which were also adjacent) had a delta of at or near 100% change: 0.3 and 0.35. These were also the only two segments that had delta values from the low engagement greater than the absolute value of the inverted difference (where low/normal engagement occurred at a higher rate than high engagement).

Further, we extracted three sub-categories of the data to determine if there were any differences between sessions of differing lengths. For this, we extracted sessions with 1-10 operations (Figure 2.11), 11-25 operations (Figure 2.12), and 26+ operations (Figure 2.13). These groups represent 24.7%, 20.4%, and 54.8% of the total sessions, respectively.

For the 1-10 operation group, we see a somewhat similar pattern to the full operation population, with a higher occurrence of high engagement operations at 25% (3rd highest),

30% (highest), 35,% (2nd highest), 90%, and 100% of the session length. Interestingly, there is a significantly higher relative instance of low engagement operations from 45% to 85%.

For the 11-25 operation group, there are some interesting deviations of relative high engagement from the combined version. Specifically, there is a clear grouping around 15% in the session, as well as the highest relative incidence at 60% in the session. Overall, This subgroup has relative high engagement at 10%, 15% (3rd highest), 20%, 55%, 60% (highest), 65%, 70%, 75%, 95%, and 100% (2nd highest), with bell-shaped curves around 15% and 60%, and a lead-up curve to 100%.

Finally, for the largest group (54.8%) of sessions with 26 or greater operations, we see a clear ramp up of relative high engagement starting at 65% and continues (with interspersed slight dips) up to the termination of the session. For this rather large group, we see higher relative high engagement at 25%, 45%, 55%, 65%, 70%, 75%, 80%, 85% (2nd highest), 90% (3rd highest), 95%, and 100% (highest).

It is interesting to note the differences between each of these three groups, and how the sections of relative high engagement combine to the results seen in the overall population behavior. Additionally, from this breakdown, it does appear that there are different behaviors in the periods of engagement that are correlated with the operations total per session. For the short sessions (1-10 operations, which also includes the 6 and 7 operation spikes seen in 2.4), there are apparent periods of engagement at around 30% in the session as well as near the end of the session, with a large dip in relative occurrence of engagement in the period between those points. The 11-25 session group sees a similar distribution as the combined group, but with a slide of high engagement to 20% earlier in the session to around 15%. The 26+ operation sessions, for the most part displays a ramp-up to high engagement as the session reaches its end. Note that this pattern is measured across sessions with operation counts of several hundred, as well.

Based on this data, and the previous session linger data, we interpret three patterns that this data appears to support:

1. The occurrence of successive instances of low then high engagement establishes a user behavior “*configure, then examine*” pattern.
2. This “*configure, then examine*” pattern represents the most common use case for the ADVANCE system.
3. Sessions of different total operation counts demonstrate different specific points of high engagement, but follow the same pattern of “*configure, then examine*”.

$$\text{Linger Time (LT)} \geq 2\sigma + \mu \tag{2.3}$$

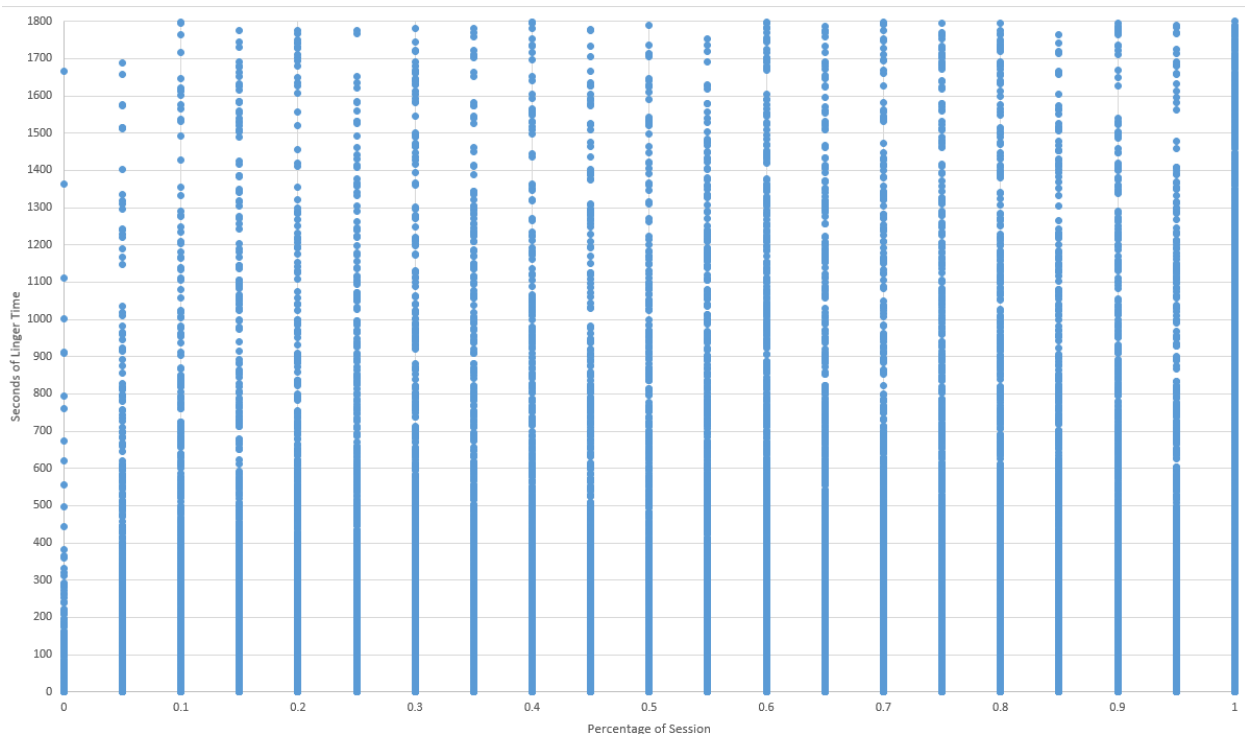


Figure 2.8: Plot of location in session with linger time of operation. Note the regular distribution of linger times throughout the portions of the session.

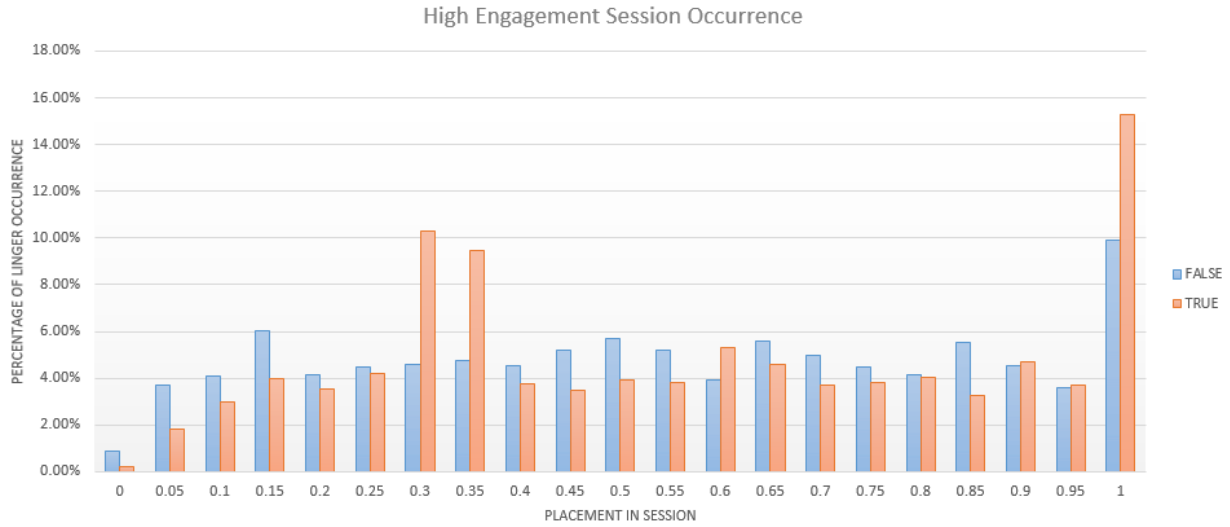


Figure 2.9: Chart of high engagement operations, listed by placement in the total session length.

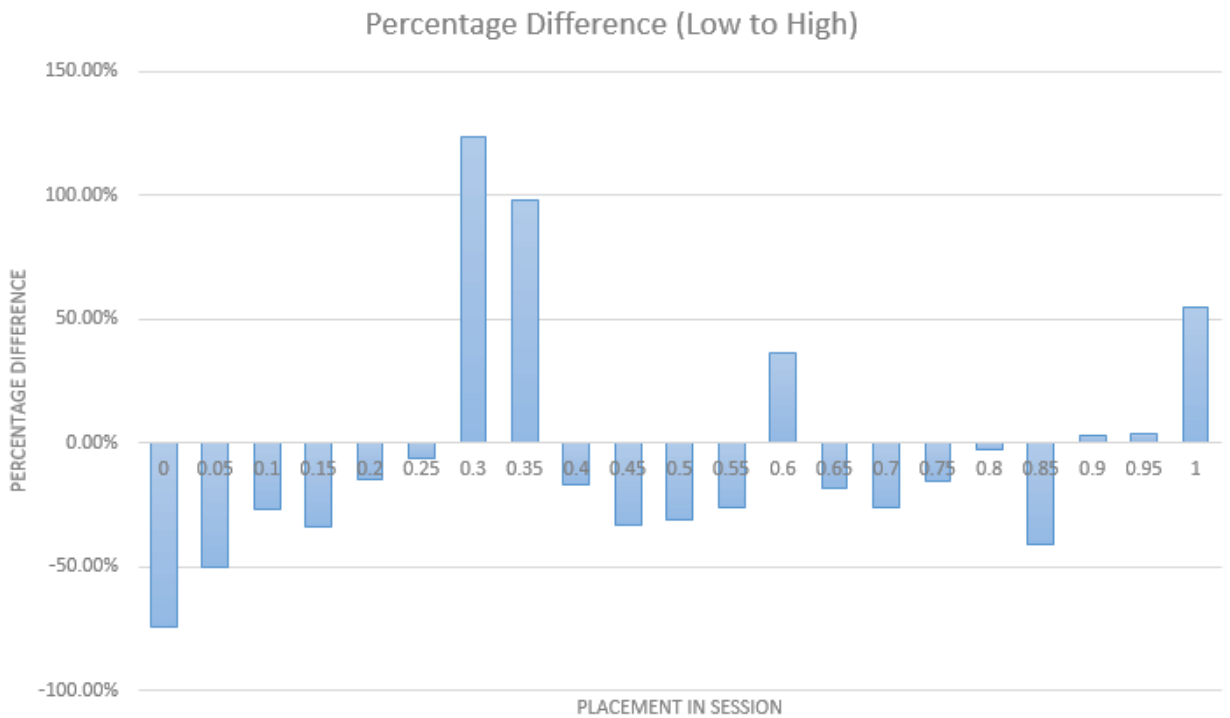


Figure 2.10: Chart of measured difference in low and high engagement. Positive percentages indicate that high engagement occurred at a higher comparative rate than low engagement at that point.

2.4.3 RQ3: Age and Duration

With our third and final question: “Are there any patterns in the dataset analysis time parameters that might imply a preference or a possible motivation to modify the default?”,

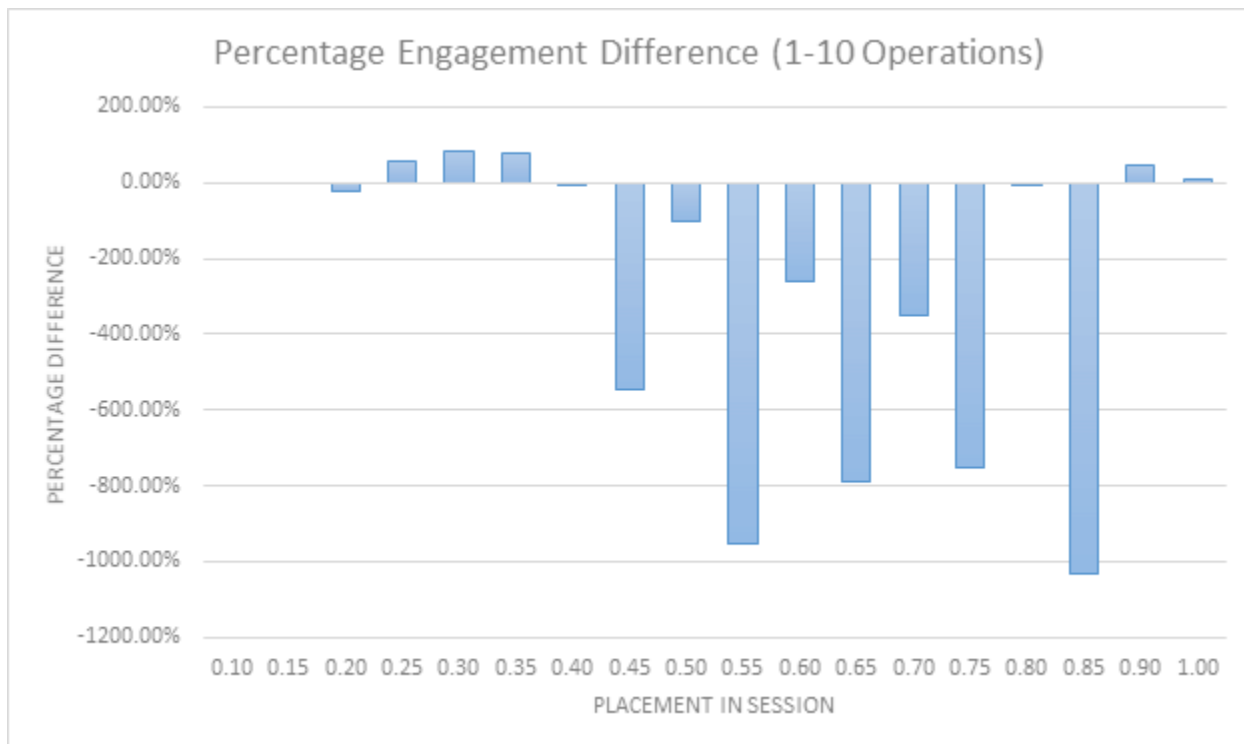


Figure 2.11: Chart of high engagement operations for sessions with 1-10 operations, listed by placement in the total session length.

we examine the extracted meta-parameters of Age and Duration, including any bias effect that their default values may introduce to the use of the system.

Age/Duration Relationship Given that *start date* and *end date* were among the most prevalent data points present in the vast majority of web service calls, and that a default for these values exist in most of the ADVANCE portal deployments, we wanted to examine both what usage data this created as well as any latent impacts that this default might have on analysis. From this reasoning, a calculation and plot of the intersection of the calculated values for the analyzed dataset’s Age and Duration was performed. Definitions for these variables are (where *Analysis Date* is the date that the user used the system):

$$Age = Analysis\ Date - End\ Date \tag{2.4}$$

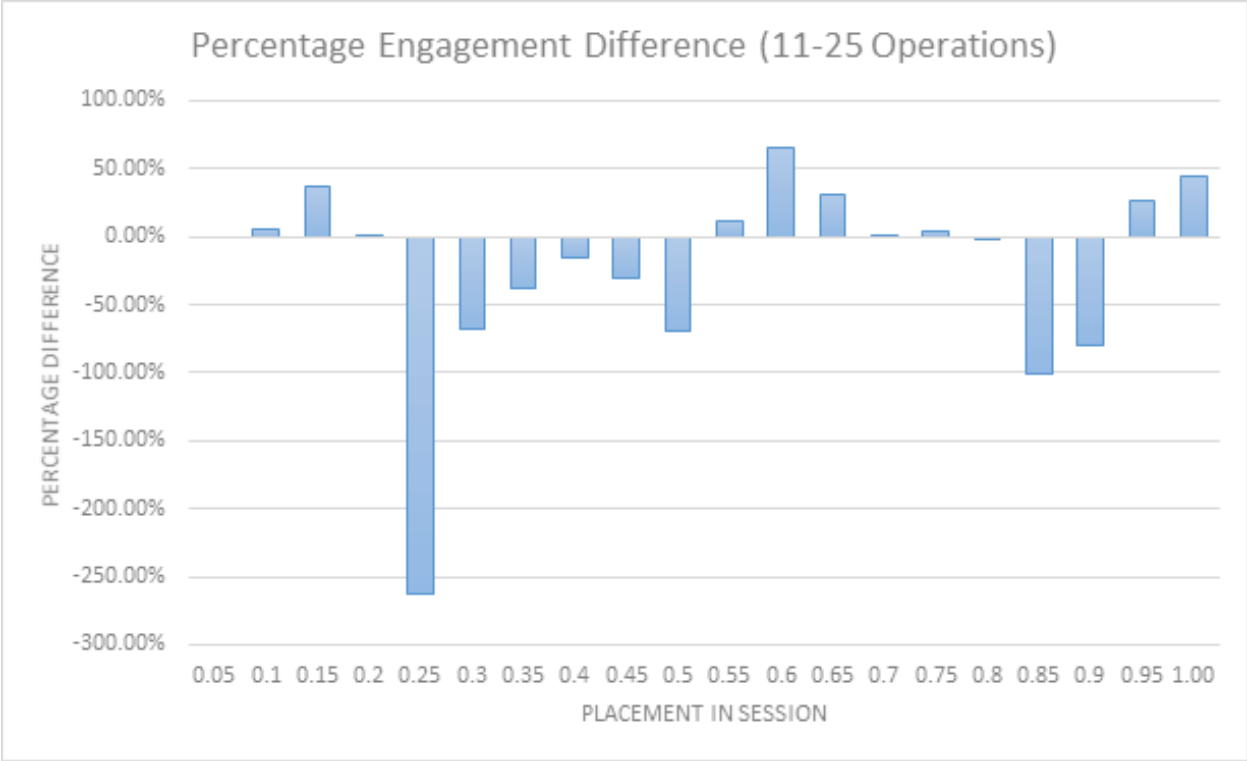


Figure 2.12: Chart of high engagement operations for sessions with 11-25 operations, listed by placement in the total session length.

$$Duration = End\ Date - Start\ Date \tag{2.5}$$

These values were extracted for each of the called datasets by maintaining a state machine with the values of both the start and end dates that were in effect for each of the calls at the time, regardless if those parameters were included or not. This was important due to the way that the logging was performed, since not every call contained a full state description.

The chart of this data can be found in Figure 2.14. The relative size of each of the points (circles) is directly related to its frequency of occurrence, demonstrating very quickly where significant clusters occur. The most prevalent intersection (indicated by the largest circle), is the confluence of Age=365 and Duration=365, where each value is measured in days. Given that this value (one year ago to date) is the default setting for many of the

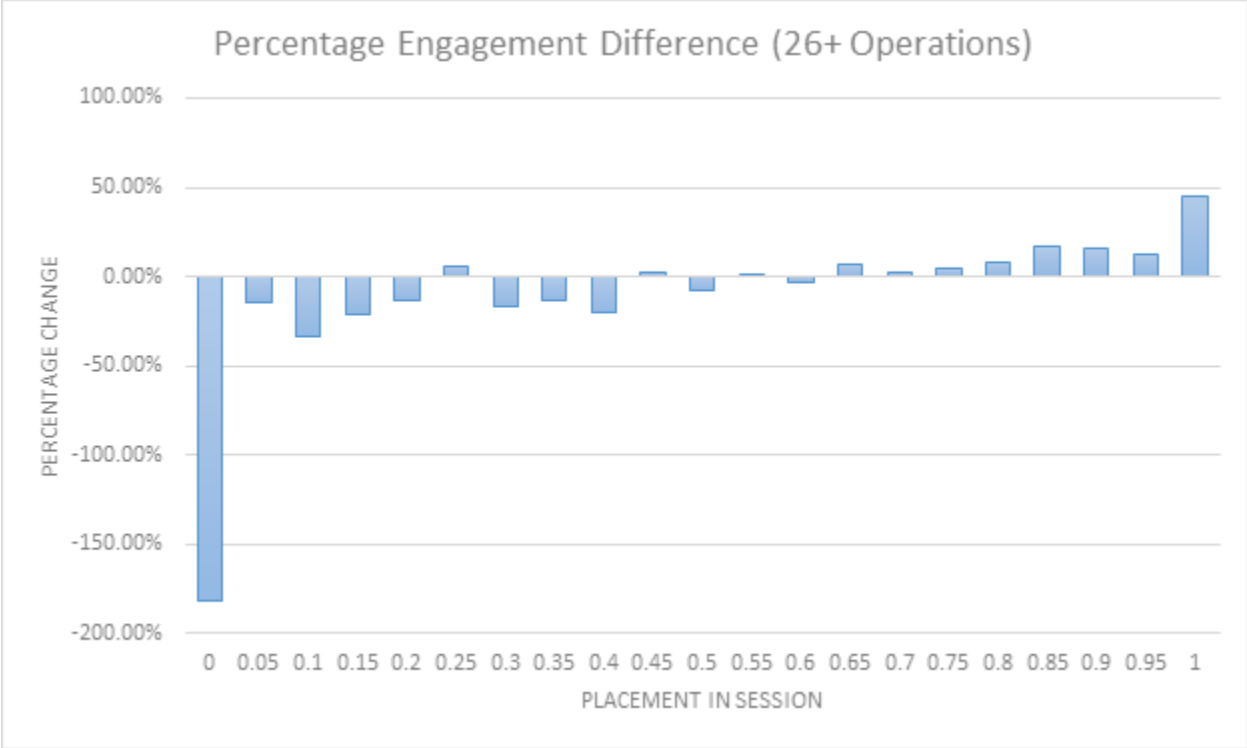


Figure 2.13: Chart of high engagement operations for sessions with 26+ operations, listed by placement in the total session length.

instances of ADVANCE and its variants, its overwhelming presence is not unexpected. In addition to this default, several other ready-made options are available to users (Shown in Table 2.2), though not all are available in every dataset.

Unfortunately, these options are user interface shortcuts to modify the start and end date values, and are not recorded as the explicit option that was chosen. As such, it is impossible to determine whether a value for the start and end dates was chosen as a result of these options, or by explicit action that happened to reflect the same values as that option. Regardless, there is still value in the recorded choices for these values, though how they were reached is not entirely clear.

Examining the chart further, it appears that many (if not most) of the large occurrence items are related to the options listed in 2.2. Several linear (horizontal, vertical, and sloped) relationships occur around multiples of standard time blocks (week (7 days), month (30 or 31 days, except February), and year (365 days)). Some non-standard obser-

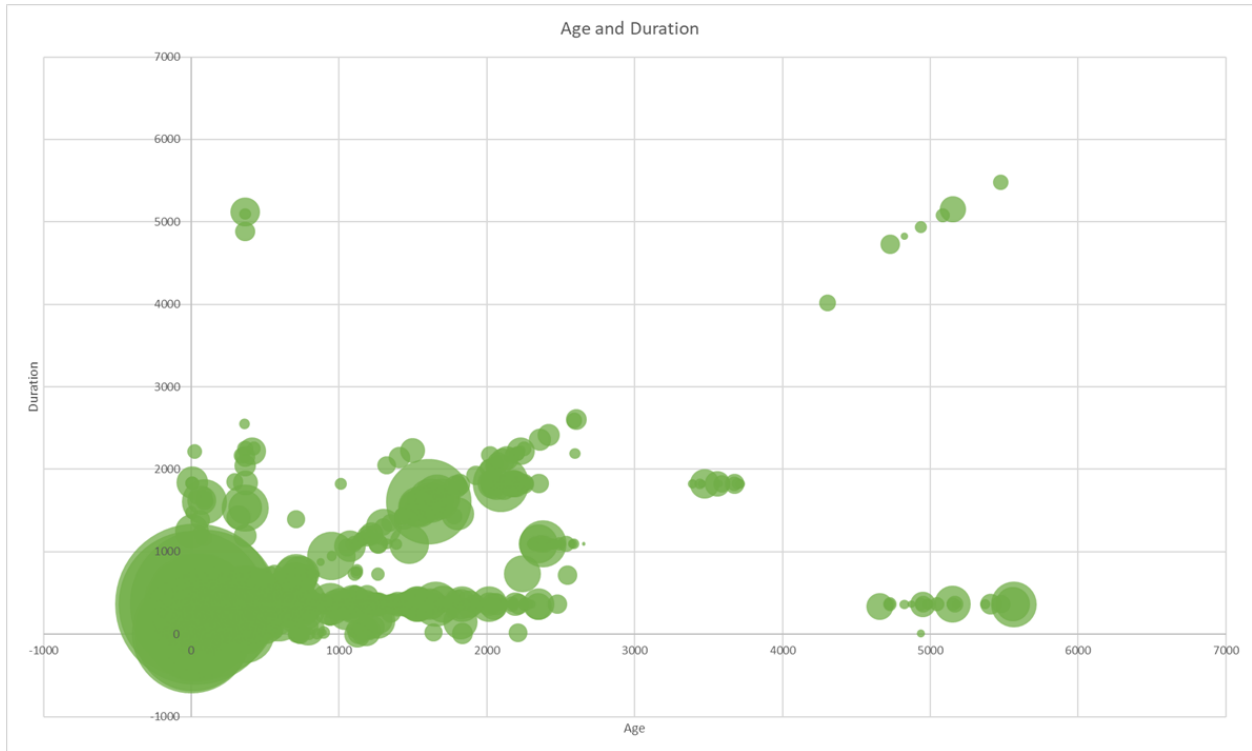


Figure 2.14: Graph of Age and Duration. Larger circles represent a larger occurrence of that Age/Duration intersection.

Table 2.2: Calculated Age and Duration values, created for the available ADVANCE default options for start date and end date.

Option	Age/Duration Values (in days)
<i>All</i>	Max Age and Duration for dataset
<i>Today</i>	Age = 0, Duration = 1
<i>Yesterday</i>	Age = 1, Duration = 1
<i>Within Last 7 Days</i>	Age = 0, Duration = 7
<i>Current Month</i>	Age = Variable, Duration = Variable
<i>Previous Month</i>	Age = Variable, Duration = 30
<i>Previous 2 Months</i>	Age = Variable, Duration = 60
<i>Within Last 30 Days</i>	Age = 0, Duration = 30
<i>Within Last 60 Days</i>	Age = 0, Duration = 60
<i>Within Last 90 Days</i>	Age = 0, Duration = 90
<i>Within Last Year</i>	Age = 0, Duration = 365
<i>Year to Date</i>	Age = 0, Duration = Variable
<i>Previous Year</i>	Age = Variable, Duration = 365
<i>Custom Date Range</i>	Variable based on range

vations include multi-year datasets up to the previous year, single-year datasets for years past, and a very prevalent (the next largest to 365/365) occurrence of 5 years ago to date. These observations are in-line with commonly used reports that are sometimes created for regular export and email (an additional feature of ADVANCE).

Based on this observation, it appears that user choices are strongly aligned with the choices provided by the date options that are available. However, it is unclear from this data as to the true preference of the default present in many of the portals.

Clustering As an additional effort to identify any latent categorizations that would apply across the user set, we investigated to see if any latent clustering was apparent in the non-categorical values (i.e. Age and Duration) within the dataset. As these values do not contain differentiable attributes, as the variable and filter values might[13], we chose a simple intersection-based clustering analysis. Based on this reasoning, a k-means analysis was completed on the x/y intersection of both age and duration. However, given the overwhelming prevalence of the default values, and the very linear tendency of the values that didn't happen to be in that category, the results of a set of clustering attempts from 2-15 clusters yielded non-useful results. While 3 clusters (shown in Figure 2.15) is reasonably divided, the 4 cluster variant (shown in Figure 2.16) is almost pure noise. This pattern of noise continues, in varying manifestations up to the 15 cluster version. Based on this result, no additional cluster numbers were attempted. However, this does show that simple clustering of users based on their Age/Duration preferences would not necessarily yield a similarity-based item recommendation that would be useful. In that regard, this does show what *not* to attempt recommendations on the basis of, and suggests that other methods should be explored instead.

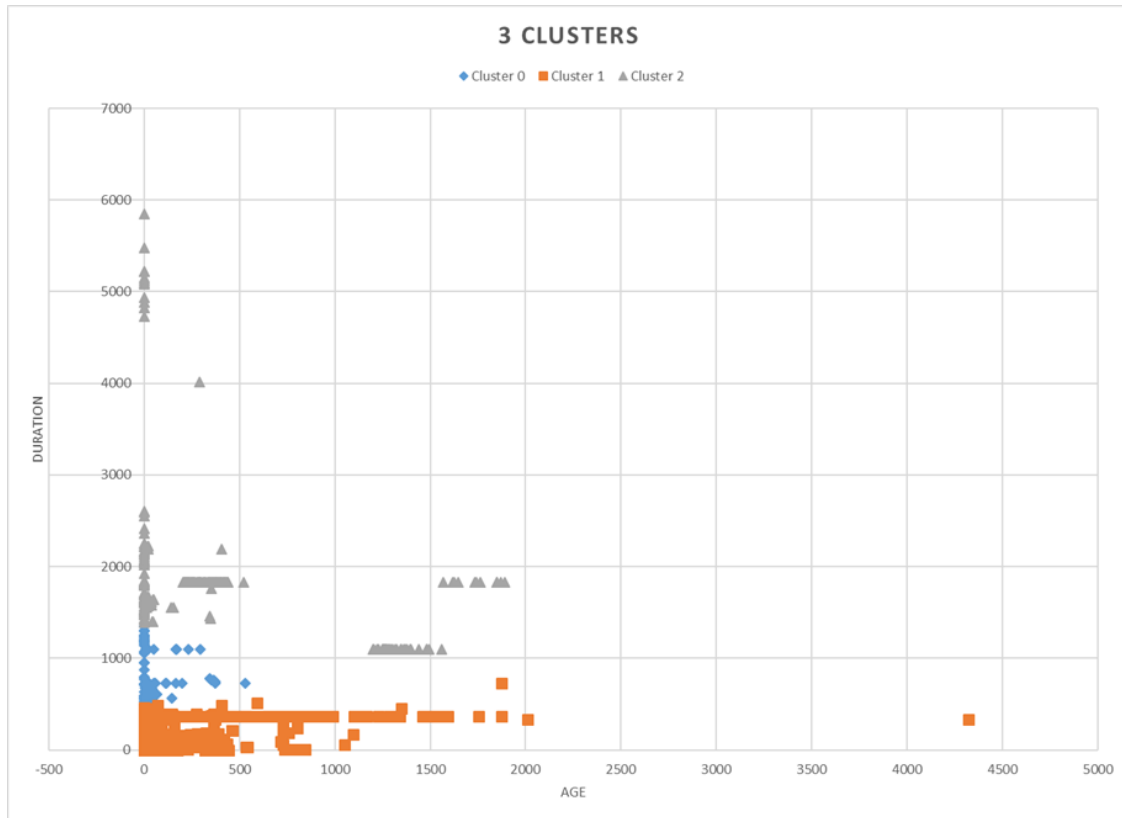


Figure 2.15: K-Means Cluster Analysis of Age and Duration - 3 clusters.

2.5 Conclusions and Future Work

Importance of “*The First Result*” Based on the analysis of the sessions, linger times, and operation characteristics, we conclude that there is strong evidence that the first configured result is likely the most important, if not the sole, item that the user significantly interacts with during each of their distinct visits to the portal. This is evidenced by the behavior seen in sessions of <10 operations, as well as the prevalence of operations of a length that is consistent with the minimum input to achieve a distinct result from the interface. While there is evidence of subsequent investigations, and a non-trivial amount of user sessions that include long (sometimes extremely long) linger times for operations, the majority of user interactions are concentrated on logging in, finding one or two results, and departing until the next session. This pattern suggests that system design that favors this behavior would improve session efficiency (which could be measured as the ratio of high

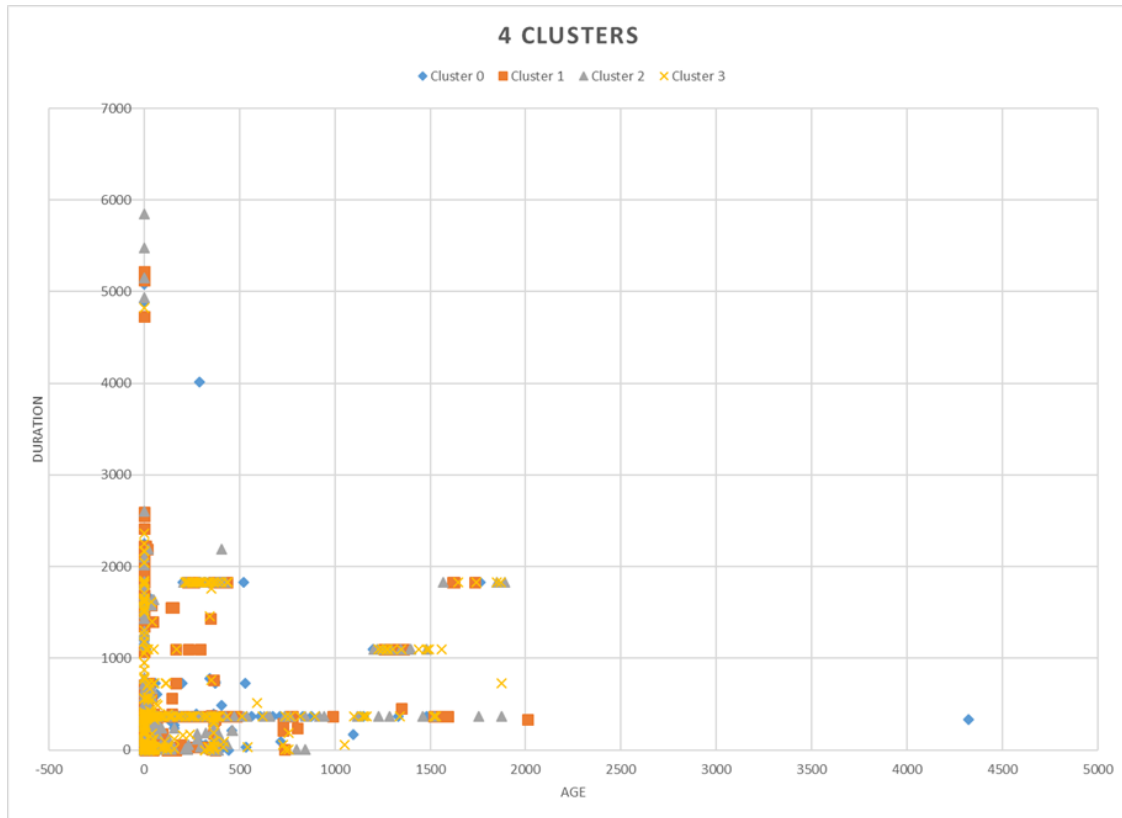


Figure 2.16: K-Means Cluster Analysis of Age and Duration - 4 clusters.

engagement operations to the session length) by reducing any "flailing" that may be occurring due to system design.

Additionally, dataset familiarity could also promote more useful sessions by removing the necessity for the user to question whether the system even has the data that they are looking for. While it is likely that meandering sessions do provide value to the user, these exploratory sessions come at an opportunity cost to other tasks, which may or may not be desirable for the users. In our experience with ADVANCE, data mining and analysis systems are useful as tools to answer questions, many of which are externally driven (due to requested reports, public data requests, or organizational fact finding) and users do not always have the time that would be necessary to acquire a general familiarity with the data. This reiterates the importance of aiding the user, wherever possible, in finding what they are looking for with the least time required.

User Behavior and Implied Intent It is understood that implying intent based on linger time isn't an ideal indicator of user preference. The lack of other data makes elucidation of true intent difficult, especially given that anecdotal evidence seems to suggest that similar activities can be performed by a user with very different intents (e.g. meandering aimlessly through the data just to be familiar looks often like an attempt to answer a specific question, but really the user is just exploring the system's capabilities). Collection of explicit feedback is possible, but (as with other systems aiming to recommend items), care must be taken to adjust to the user's willingness to contribute to the system attempting to cater the results to them. Being too overbearing in the process can be annoying to the user and act as an obstacle to building and retaining trust in the recommendation system. Conversely, the system must be constructed to trust the "right" users (ones that provide good feedback) [11].

Role of Defaults Theoretically, defaults are implemented to provide a reasonable place for the user to start an analysis. While any global default, implemented for any system, will not meet the needs or expectations of all users, it should at least be a non-harmful start state that minimizes negative influence on the user. The data collected in our research does not strongly suggest that a new set of defaults should be chosen, but an experiment with a localized exemption from the default could yield useful results. This could possibly be implemented as a blank "start state" for the user in which they choose the opening parameters for their session, perhaps including a save feature to avoid requiring modification each time. This however, reiterates the issues with defaults in systems such as these. If the changes required to overcome the defaults are relatively trivial, the user demand for their modification will likely be lessened, further conflating accurate collection of user behavior in cases where the start state isn't simply thrown out.

Future Work Given the results of this analysis, it seems important to continue to investigate methods that could be used to guide users to valuable results in the system. Con-

structuring a picture of what the user’s goal might be, as well as providing non-harmful recommendations, seems to be a promising area of research, especially within in a domain (purpose built data analytics systems) that does not seem to receive as much research attention as other areas. This observation of lower research volume, which is possibly due to the proprietary data and security concerns surrounding certain datasets, also encourages methods of data extraction and translation that would promote sharing of this type of data, albeit with any sensitive content removed.

References

- [1] AYE, T. T. Web log cleaning for mining of web usage patterns. *ICCRD2011 - 2011 3rd International Conference on Computer Research and Development 2* (2011), 490–494.
- [2] BAKER, R. S. J. D. R., AND YACEF, K. The State of Educational Data Mining in 2009 : A Review and Future Visions. *Journal of Educational Data Mining 1*, 1 (2009), 3–16.
- [3] BRACHMAN, R. J., AND ANAND, T. The Process of Knowledge Discovery in Databases: A First Sketch. *AAAI-94 Workshop on Knowledge Discovery in Databases* (1994), 1–11.
- [4] BRUSILOVSKY, P., AND MAYBURY, M. From adaptive hypermedia to the adaptive web. *Communications of the ACM 45*, 5 (2002).
- [5] DRACHSLER, H., AND GRELLER, W. The pulse of learning analytics understandings and expectations from the stakeholders. *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge - LAK '12*, May (2012), 120.
- [6] GONÇALVES, B., AND RAMASCO, J. J. Human dynamics revealed through Web analytics. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics 78*, 2 (2008).
- [7] HE, D., AND HARPER, D. Detecting session boundaries from Web user logs. *Proceedings of the BCS-IRSG 22nd Annual Colloquium on Information Retrieval Research* (2000), 57–66.
- [8] HOFGESANG, P. Relevance of time spent on web pages. *In Proc. of WebKDD 2006: KDD Workshop on Web Mining and Web Usage Analysis, in conjunction with the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006)* (2006).

- [9] JANSEN, B. J., SPINK, A., AND BLAKELY, C. Defining a Session on Web Search Engines. 862–871.
- [10] KELLAR, M., WATTERS, C., DUFFY, J., AND SHEPHERD, M. Effect of task on time spent reading as an implicit measure of interest. *Proceedings of the American Society for Information Science and Technology* 41, 1 (sep 2005), 168–175.
- [11] O'DONOVAN, J., AND SMYTH, B. Trust in recommender systems. In *Proceedings of the 10th international conference on Intelligent user interfaces - IUI '05* (2005).
- [12] PARRISH, A., DIXON, B., CORDES, D., VRBSKY, S., AND BROWN, D. CARE: an automobile crash data analysis tool. *Computer* 36, 6 (jun 2003), 22–30.
- [13] SHEPITSEN, A., GEMMELL, J., MOBASHER, B., AND BURKE, R. Personalized recommendation in social tagging systems using hierarchical clustering. *Proceedings of the 2008 ACM conference on Recommender systems - RecSys '08* (2008), 259.
- [14] SIEMENS, G. Learning Analytics : Envisioning a Research Discipline and a Domain of Practice. 4–8.
- [15] SMITH, R. K., GRAETTINGER, A. J., KEITH, K., AND PARRISH, A. Identifying High Frequency Crash Locations: Empowering End-Users with GIS Capabilities. *ITE Journal* 77, 1 (2007), 22–27.
- [16] SRIVASTAVA, JAIDEEP; COOLEY, R; DESHPANDE, M; TAN, P. N. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations* 1, 2 (2000), 12–23.
- [17] STEIL, D. A., PATE, J. R., KRAFT, N. A., SMITH, R. K., DIXON, B., DING, L., AND PARRISH, A. Patrol routing expression, execution, evaluation, and engagement. *IEEE Transactions on Intelligent Transportation Systems* 12, 1 (2011), 58–72.

CHAPTER 3

ARTICLE 2 - FAST (RE)INTRODUCTION TO ANALYTICS DATASETS: NOVEL AND POPULIST NAVIGATION USING USER SOURCED PATHS

3.1 Introduction

Within the research areas involving item recommendations and recommender systems in general, the primary goal is usually to adapt the recommended items solely to the preferences of the user. This goal, and the methods that support it, are extremely relevant and are very adept at accomplishing that goal. However, this goal is not an optimal goal in all use cases. Some circumstances and target systems may have preferred user patterns that are not as disparate and personal as the majority of recommender systems.

In our case, we have identified two specific goals within this deviating set of cases, primarily with the intent to promote system education through targeted recommendations that optimize those goals. These identified goals are:

1. To recommend items that promote exploration of the uncharted areas of the system.

We call this **Novel Navigation**.

2. To recommend items that promote exploration of the most popular (but not visited by the user) items in the system. We call this **Populist Navigation**.

Although our goal is to provide a mechanism to drive users to items that fit an overall system goal (not unlike some possible meta-goals held by commercial recommender systems to drive customers to certain items, regardless of preference), we do take into account

at each point of the user’s navigation through the system which items are most relevant to either goal (novel or populist), and make an effort to avoid a simple and repetitive barrage of the same items. This analysis of potential benefit is performed using a simulated replay of site activity with the existing data, which is also used in the construction of the input model (albeit a different subset of the data). This context aware approach seeks to promote system education while also taking the user’s current actions into account.

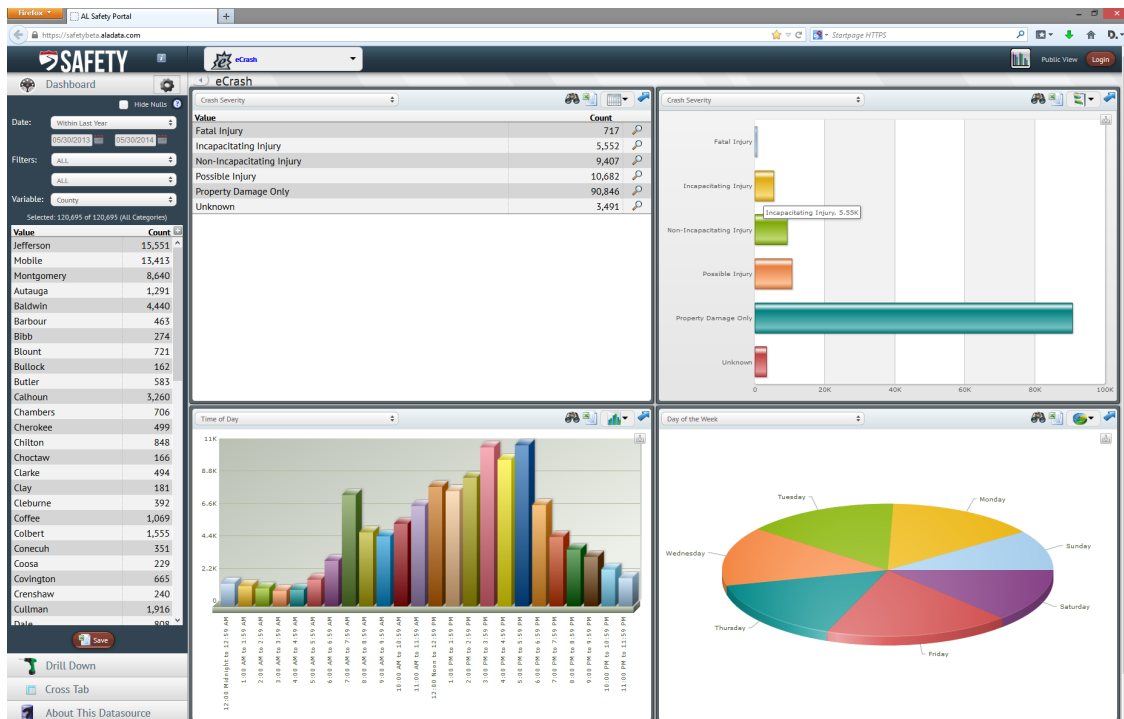


Figure 3.1: Screenshot of *ADVANCE*. The *dataset filters* are displayed in the left column and the *dataset display preferences* are displayed in the four large panels to the right.

3.2 Related Research

There are several recommender system areas that apply to the goals of users of data analytics systems. In some cases, these systems are often used in an exploratory fashion by both novice and expert users wishing to either become familiar (or refamiliar) with the patterns of usage in a large and complex dataset. This type of activity has roots in both Technology Enhanced Learning (TEL)[11, 20] as well as playlist style recommendation [2,

8]. The latter is especially unique in that it doesn't seek to find the best overall individual item, but rather a set which would be most useful to the user. In this case, both the items and the order are important.

Other areas which are also relevant include web usage monitoring [5, 10, 13] and context aware recommender systems [9, 18, 12]. Web-based systems (as many of these recommender systems are) can provide a lot of metadata that can be used to further refine the item suggestion process. Being context-aware, especially in a domain with well defined user roles (including hybrid roles), can be a primary selection factor for recommending an item (or items). Considering users as customers in the system, but slightly altering the process so that the goal is less to make a sale than provide relevant items with no optimal outcome, e-commerce approaches to recommendations[15] also emerge as interesting sources of approach methods.

A data discovery and analytics tool is an obvious choice for applying data mining, statistical analysis, or machine learning methods to extract useful patterns from the data in a recommender system. Many times, there are non-obvious data relationships that users might not think to investigate on their own, but when presented with suggestions drawn from even the most elementary of mining methods, the user could gain some benefit in looking at the data in a different way. Complicated datasets lend themselves well to this type of assistance [6], and given the various types of data types seen in *ADVANCE* (e.g. vehicle crash and traffic citation data), the impact of providing a more targeted and accurate presentation of the data could be substantial.

Markov chains are also not uncommon in analysis for recommender systems, with use for personalized news[7], link prediction[22], and adaptive web navigation[1].

3.3 Data Collection and Path Construction

In *ADVANCE* [16], there are several state and local agencies that have either their own public safety dataset (e.g. vehicle crashes) or a selective view into the larger dataset

based on their location or organization membership. For example, if the user is a member of Agency A, then members of Agency B cannot see data specific to that area, and vice-versa. In most cases, this is either a data ownership decision or an explicit request from the agencies involved.

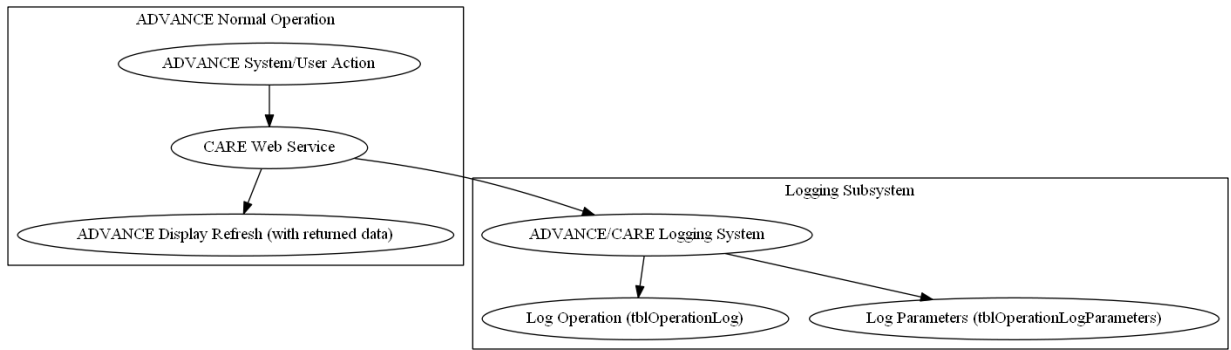


Figure 3.2: Diagram of the logging process for ADVANCE.

The diagram, shown in 3.2 shows the process for logging the activity in ADVANCE. The CARE web service calls are sent to the logging subsystem, which logs the method called as well as all of the key/value parameters. The primary variables of interest, that are easily discernible as inputs from the main page of ADVANCE:

- Start Date
- End Date
- Filter
- Filter Variable
- Chart 1-4 Variable
- Chart 1-4 Type (bar, pie, etc.)

Coincident with the migration of the CARE engine from major version 9 to major version 10, a diagnostic logging process was inserted to capture activity sent to the CARE

web services. The results of this logging process were captured in two database tables: *tblOperationLog* and *tblOperationLogParameters*, each described in Tables 3.1 and 3.2.

Approximately two-and-a-half years of log data was collected as part of system telemetry and for debugging purposes (i.e. to fix programming errors). There are approximately 14,000,000 records in the *tblOperationLog* table and 19,000,000 records in the *tblOperationLogParameters* table.

Within this data, there are two primary categories of collected data: *dataset filters* and *dataset display preferences*. Dataset filters refers to parameters such as start and end date, predefined filters (built manually by the creators of the datasets), and one or more filtering criteria (chosen from the available categorical variables of the dataset). These are shown in the left pane of Figure 3.1. The dataset display preferences, shown in the four large panels of Figure 3.1, are composed of the four available variable display tiles and the different display methods (graph types) that can be used to visualize that data. Once the user has used the filters to pare down the dataset, he can then explore the frequencies of up to four unique variables in that set.

Table 3.1: User operation and context database table - *tblOperationLog*

Name	Type	Description
<i>id</i>	integer	Identity (auto-incrementing) operation ID
<i>datetimeOccured</i>	datetime	Date and time of operation call.
<i>logSessionId</i>	varchar(50)	Unique session identifier
<i>username</i>	datetime	Obfuscated (hashed) username
<i>userOperation</i>	varchar(50)	CARE web service method that was called
<i>portalName</i>	varchar(75)	Website that was using the web service (this allows identification as to whether it was ADVANCE or one of the other variant portals)

The *logSessionId* (listed in Table 3.1) was created shortly following the commencement of the logging operation to improve session tracking accuracy. Prior to the implementation of this explicit field, a process to identify sessions was manually implemented in code that identified user activities as being part of the same session if the duration between each ac-

Table 3.2: User operation parameters table - *tblOperationLogParameters*

Name	Type	Description
<i>id</i>	integer	Identity (auto-incrementing) parameter list ID
<i>OperationLogId</i>	integer (foreign key)	Reference key to the parent operation
<i>parameterName</i>	varchar(50)	Method call parameter name (dynamic, free text)
<i>parameterValue</i>	varchar(100)	Method call parameter value (dynamic, free text)

tivity did not exceed the user login timeout. This is a fairly reliable method, since if the user had no logged activity, they would have been automatically logged out and a new session would have been detected because of the break in activity. However, this explicit field was put into place to avoid false session breaks, since it was now tied to the authentication subsystem (i.e. it is aware when the user is logged into the same contiguous session).

The unique session identifier, due to infrequent errors in the unique GUID (globally unique identifier), will sometimes result in an empty value for the session. Although this does not cause any issues with our extraction, given that we iterate over the user activity that is contiguous within the session, other methods that we might employ in future work may be affected by this occurrence. Given that some extraction methods are not user-centric, and instead focus on sessions individually, these sessions may need to be removed from the candidate dataset.

Following the collection process, this logging data was used to create a Markov chain [14] of each of the changes to the *start state of the session* (a selection of which is shown in Figure 3.3). A state machine of the current state, as well as each delta from that operation, is kept and then used to calculate a global graph with the vertices and links. Additionally, we construct this chain both as a global set containing all of the changes in all of the different datasets, and separated by the specific website (e.g. *ADVANCE* portal) from which the user accessed the data. In many cases, the same datasets are available through multiple portals, but access to each is determined by membership in a specific agency. We

separate the feedback by this criteria due to the implicit differences in goals that each agency has for the data and to aid in identification of any portal specific patterns that may be present. Being sensitive to the unwanted effects that power users may have[21], we filtered certain internal users' results from the process to avoid bias introduced by test and debugging behaviors.

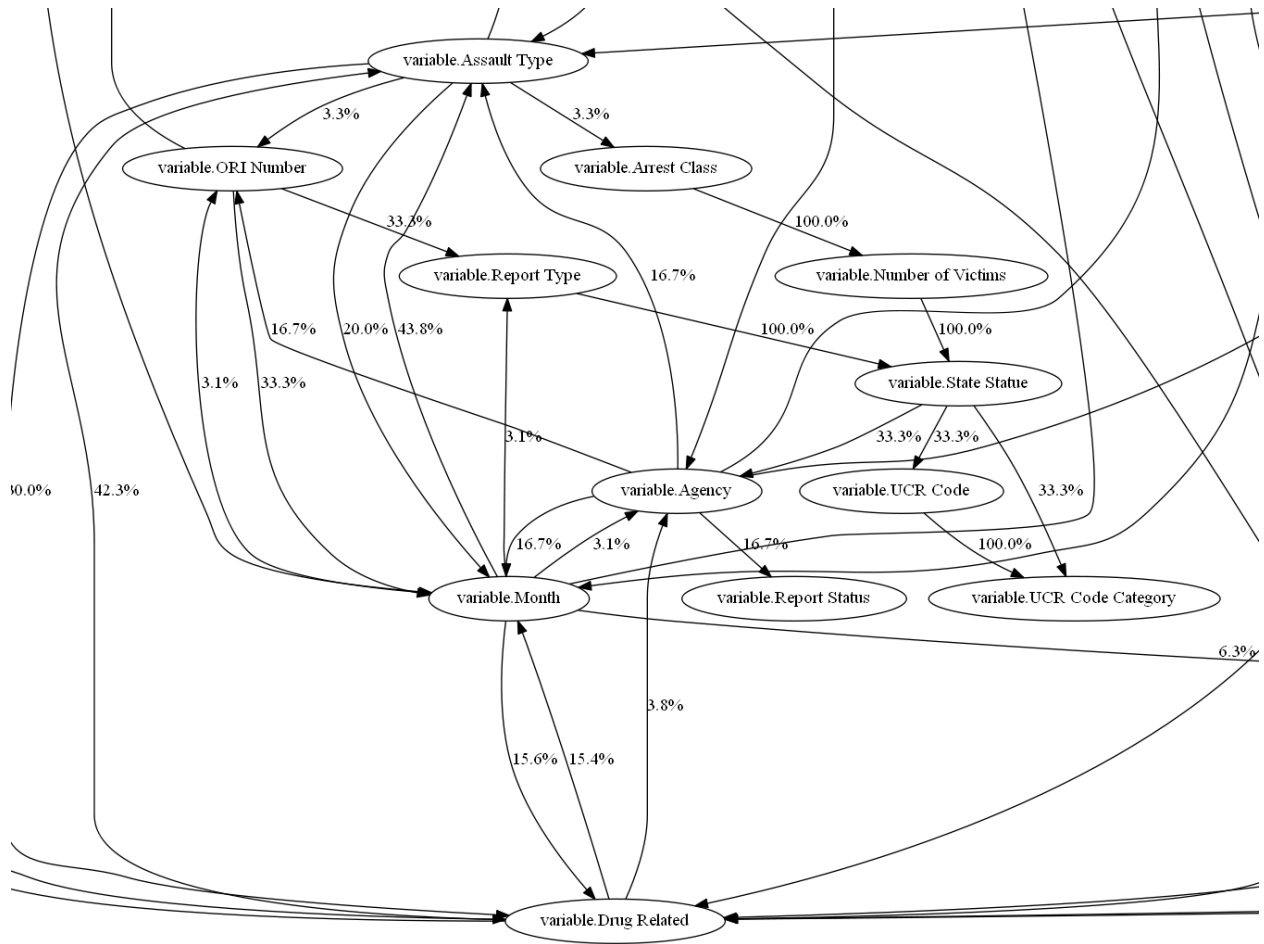


Figure 3.3: Portion of the Markov chain describing the state transitions between variables.

3.4 Recommendation Methodology

For recommendations based on user to user similarity, the use of collaborative filtering as a tool to guide users to items that they would prefer maintains the core method of determining user preferences and then using preference matrices to drive recommen-

dations. This is based on the assumption that the system is intended to elucidate (and then drive the user toward) a subset of the available items best suited to their preferences. However, in our case, the system is intended to somewhat normalize the user’s choice of path. Within this system, the user is (or should be) acting in a role that prefers alignment with organizational goals than strict personal preference. This does not mean diminish individual system use behavior and preferences, but only to highlight the adjusted goal that does not necessarily optimize toward an individual’s singular choices.

This deviation from the typical recommendation paradigm implies a system that extracts and distills the population’s overall behavior into a set of navigation instructions. While the recommendation overall will be driving the user toward a central tendency of behavior (in some cases), the individual recommendations will be provided in the context of the user’s current location in a session. Our approach uses simulated responses taken from the same superset of data used to construct the recommendations, and is subject to some potential sampling bias and assumption of user response (since the recommendations were never presented to the users, nor any subsequent responses collected).

Given that one of the primary goals is to introduce (or reintroduce) users to a new data source, construction of one or more summary Markov chains provides a mechanism to navigate the user’s session while (in parallel) using the link weights to give the user navigation advice. With this in mind, there are two primary methodologies we used for promoting user engagement in the system: Novel Navigation and Populist Navigation.

3.4.1 Novel Navigation

In cases where a user has extensive experience with the domain or dataset, and has navigated the system frequently, providing recommendations that promote popular items within the system is not necessarily the method that contributes the most value. As evidenced by the user paths in the data, there are portions of the data that often go unvis-

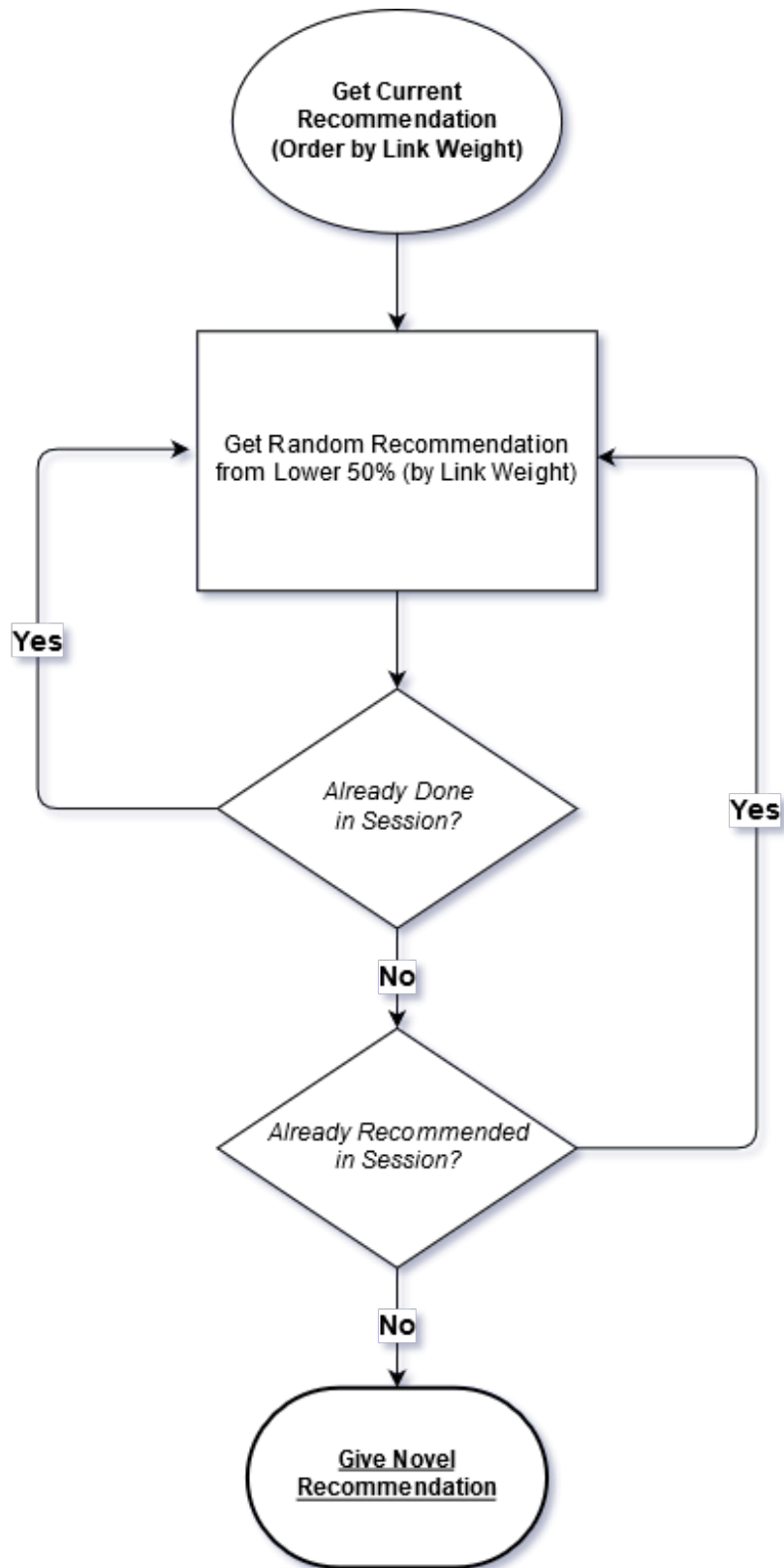


Figure 3.4: Recommendation construction process for Novel Navigation.

ited by most users. These lesser explored areas of the system may contain latent information that users either don't recognize or understand any value they may provide.

Given that, to appear as a node in the Markov chain, the node must have been visited at least once, this does not promote items that may be truly novel (i.e. have no user exposure *at all*). However, for any nodes that have been visited *at least once*, the system captures any visit data to or from that node and can present it as a recommendation to visit.

For the process of providing novel recommendations, a *modified least visited* method was used. For this method, a recommendation set is calculated at the current node, with a random node being chosen from *the lower 50%* of that set. This promotes novel exploration across the spectrum of lesser visited nodes by spreading the choice exploration among all previously visited out-nodes from the current location. By keeping the candidate set at the under 50% mark, as well as randomizing the recommendation, we are able to avoid having long term use of the system creating a bottom-heavy populist navigation, in which the least visited nodes are so overwhelmingly recommended that those nodes become those that are recommended during a non-novel recommendation scheme. This process is described graphically in Figure 3.4.

3.4.2 Populist Navigation

A common problem experienced among new staff that are responsible for understanding an organization's data is how to most efficiently acquire a basic understanding of that data. These roles can often include not only simple understanding of the data but also involve processing, transformation, and production of summarized reports and decisions.

In many cases, adequate time is not given between change of personnel to allow for a proper transfer of knowledge, leaving the analyst to guess at the best methods by which to both quickly understand and make actionable the data at their disposal. Specifically, in the case of users of systems like ADVANCE, the best options are to investigate the sys-

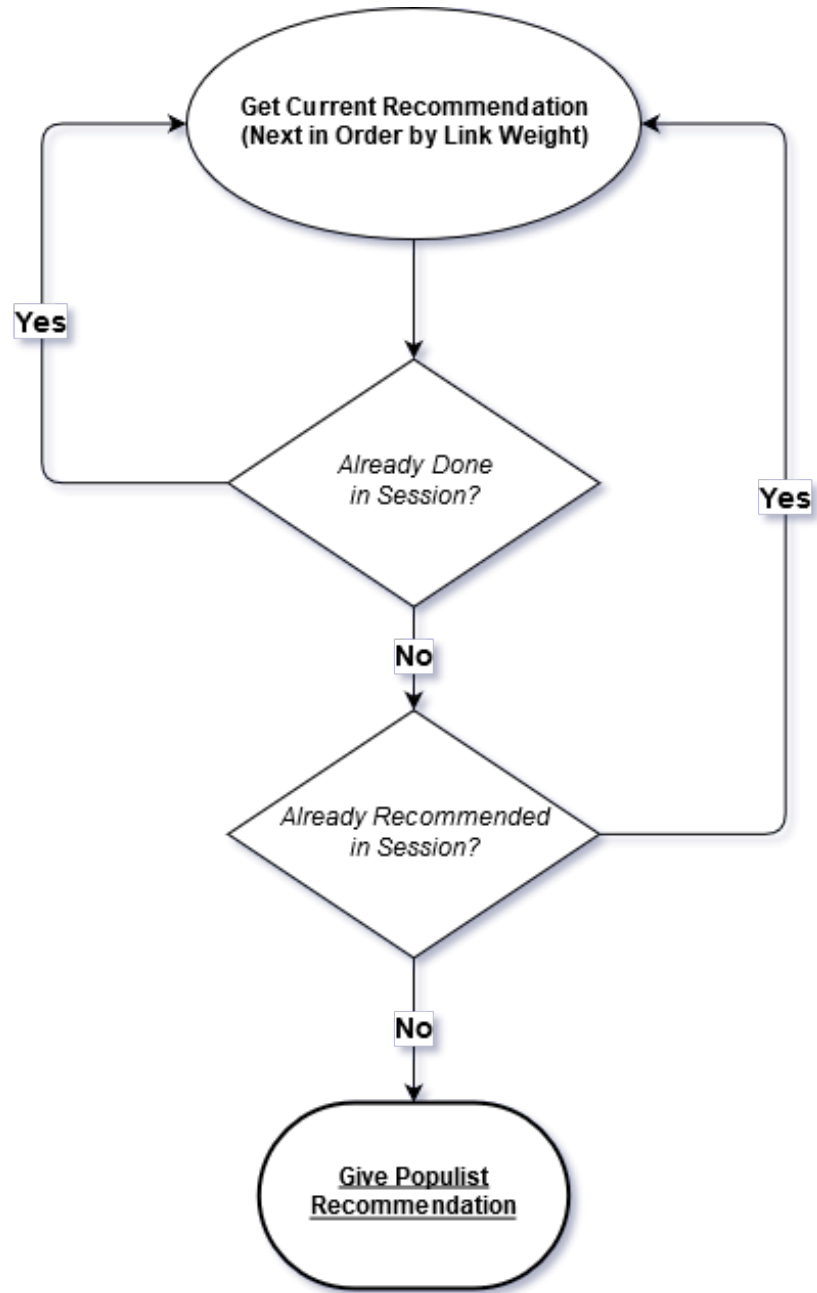


Figure 3.5: Recommendation construction process for Populist Navigation.

tem’s automatically produced reports, which were likely created by the previous user in that role. However, knowledge of the reports and their contents (which are often only subsets of the available data) is not of the same value as familiarity with the dataset(s) themselves.

Using path information provided by other users of the system, new users can be ”brought up to speed” on the typical variables and display methods used by other users. However, it is important in providing recommendations to not blindly guide the user to the most popular option, as this may already be a node that the user has visited. It is observed that there can be a *popularity bias*[17] that may serve to reinforce engagement in the already popular items, even if they aren’t necessarily providing feedback on the relevance of a given recommendation.

In our method, we use a *path aware populist* recommendation, meaning that we take the list of out nodes (listed in descending order by score) and cross-reference our recommendation with that of the user’s path in the session. If the user has visited the node we would recommend, we move to the next node until we either yield an out node that the user has not yet visited or exhaust the candidate list of nodes, in which case we recommend a random node from the set. This process is described graphically in Figure 3.5.

3.5 Recommendation Setup and Experiment

The choice of the granularity of the constructed Markov chains, through the use of query filters of the data, changes the types of resulting recommendations. In this case, we extracted two separate types of chains: *combined* and *portal*. Within these two sets, we divided the operations according to their session membership, evaluating each user’s activity as a complete segment for input into the construction of the weighted paths within the chain.

Furthermore, for the actual *combined* Markov chain construction, the entire set of user visit patterns were used. This full set was considered because of the similarity in domain

of the datasets between the separate portals (i.e. public safety) as well as the overlap of the specific variable names. This similarity is owed to the consistent input data used to construct the datasets, which is the result of having many of the input fields being produced by the same software system. Initially, we had considered limiting items that were globally present in all chains, but observations of subsets of the data did not exhibit discernible outlier members that did not have similar items, but only of a slightly different name. For this reason, we considered the variable membership of the datasets acceptable for use in the experiment.

With the *portal* Markov chain, each separate portal (identified by either separate target entity, authentication system, or top level domain) was constructed, for a total of four portals and corresponding sub-graphs. This set was constructed to concentrate distinctions between each set of users, often working within the same entity, within separate containers. This also has the result of emphasizing any agency specific categorical variables, especially those that are less visited. This is important primarily within the Novel Navigation recommendation scheme, as the system could possibly recommend an item that does not directly correspond to certain outlier variables. Because of the consistency of the data input, this was considered an acceptable consequence, given that a secondary portal-specific chain was being constructed which would eliminate this risk.

Using the previously constructed Markov chains (both global and portal specific), we constructed a recommender system which operates as follows:

1. For each run of the experiment, 25% of the total operations are extracted for exclusion from the construction of the Markov chain. This is done by randomly ordering the users, then including sessions (adding the total operations at a time) into the exclusion set until the 25% operation mark is reached. This also results in a random set of users included in each run of the experiment.
2. We extract the list of operations from the group of all of the current user's sessions, processing each session sequentially.

3. For each operation, the Novel and Populist sub-routines (described in Figure 3.4 and Figure 3.5, respectively) are run on the operation. For recommendations that either are operations that the user has already done or already recommended, the item is logged but not presented. Possible outcomes are described in more detail below.
4. At the conclusion of the iteration of all users and sessions, a summary output file is produced which records the recommendation results, placement of each result in the session, and the total operations per session.

During the recommendation process, checks are made to prevent repeated duplicate recommendations during a session. While the recommendation may be relevant, we wanted to prevent potentially bombarding the user with the same item over and over. Regardless, we still wanted to capture when a recommendation was considered, even if it wasn't included as a candidate for presentation to the user. With this in mind, we record these events in one of the following categories:

1. **Already Recommended** - At the time of the recommendation (in the session), the operation was already recommended. This recommendation is not presented.
2. **Already Done** - At the time of the recommendation (in the session), the operation was already in the set of operations performed by the user. This recommendation is not presented.
3. **Recommendation Given** - Non-repeated and non-performed recommendation. This recommendation is presented.

Although each specific run of the experiment does result in some variation (in the specific membership of the operation and user sets, for example), we will focus on a single specific run of the experiment. For this run, we find **4,945 vertices** and **33,845 links** in the completed global (combined) Markov chain.

Table 3.3: Operation categories, as extracted to the Markov chain.

Category	Description
<i>Chart</i>	Includes modifications of chart type (bar, graph, etc.)
<i>Data Source</i>	Modification of data source (typically refers to different forms or documents)
<i>Date</i>	Includes start and end date modification
<i>Filter</i>	Modification of any filter or filter set
<i>Hide Null Values</i>	UI command to remove null values from the list of returned values
<i>Other</i>	Various UI commands that are either specific to a portal or of minimal impact to the session
<i>Variable</i>	Includes all variable changes (primarily in the chart area)

Table 3.4: Markov chain construction results.

Granularity (name)	Vertices	Links
<i>Global (Combined)</i>	4,945	33,845
<i>Portal 1</i>	874	4,870
<i>Portal 2</i>	963	3,977
<i>Portal 3</i>	3,591	21,182
<i>Portal 4</i>	1,911	9,015

In order to reduce noise in the analysis of the results, we extracted seven major categories that represent the significant groupings of the types of values that we observed within the Markov chain vertices. These categories are (listed alphabetically): *Chart*, *Data Source*, *Date*, *Filter*, *Hide Null Values*, *Other*, and *Variable*. Descriptions of these fields can be found in Table 3.3.

3.6 Results

Before we discuss the results of the experiment, it is important to reiterate that these results are based on replay of simulated user sessions, and not the result of placing the recommendations in front of a separate user group for evaluation. This distinction is critical to note to avoid confusion and to give context for the results discussed below. In short, when we state that these recommendations were “given” and “taken”, this specifically

refers to the examination of user behavior with the background knowledge related to the recommendations that our system would have provided them during their normal session.

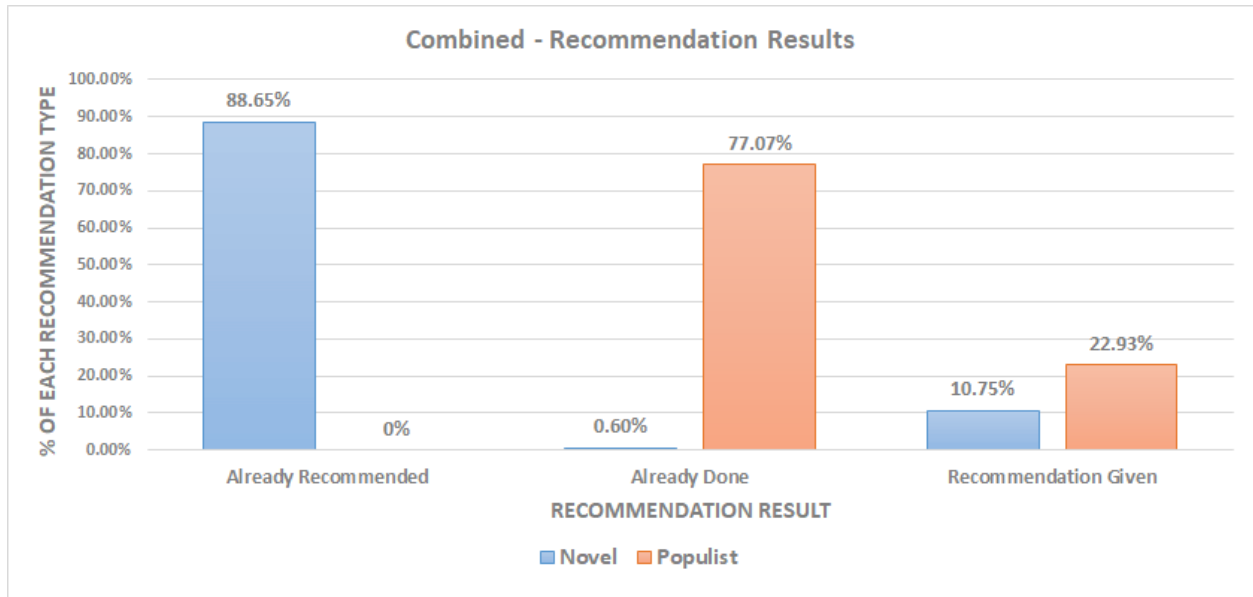


Figure 3.6: Combined - Recommendation results by action category (*Already Done*, *Already Recommended*, and *Recommendation Given*), shown as a percentage of the total calculated recommendations.

Examining the results of the recommendation simulation, we found several interesting patterns in both the Novel and Populist recommendation scenarios. First, we looked at ratio of total recommendations that the user was presented (ones in the *Recommendation Given* category) to the instances of that recommendation appearing *somewhere* in the user’s session operations. We call this ratio the *take rate*.

As previously mentioned, we tracked the individual portal performance (for user sessions that we could definitively identify as specific to one portal) for the recommendation simulation as well. For clarity (and to make clear the relative impact that each portal’s data has on any portal specific analysis), we present the total population of recommendations, broken down by portal membership in Figure 3.7

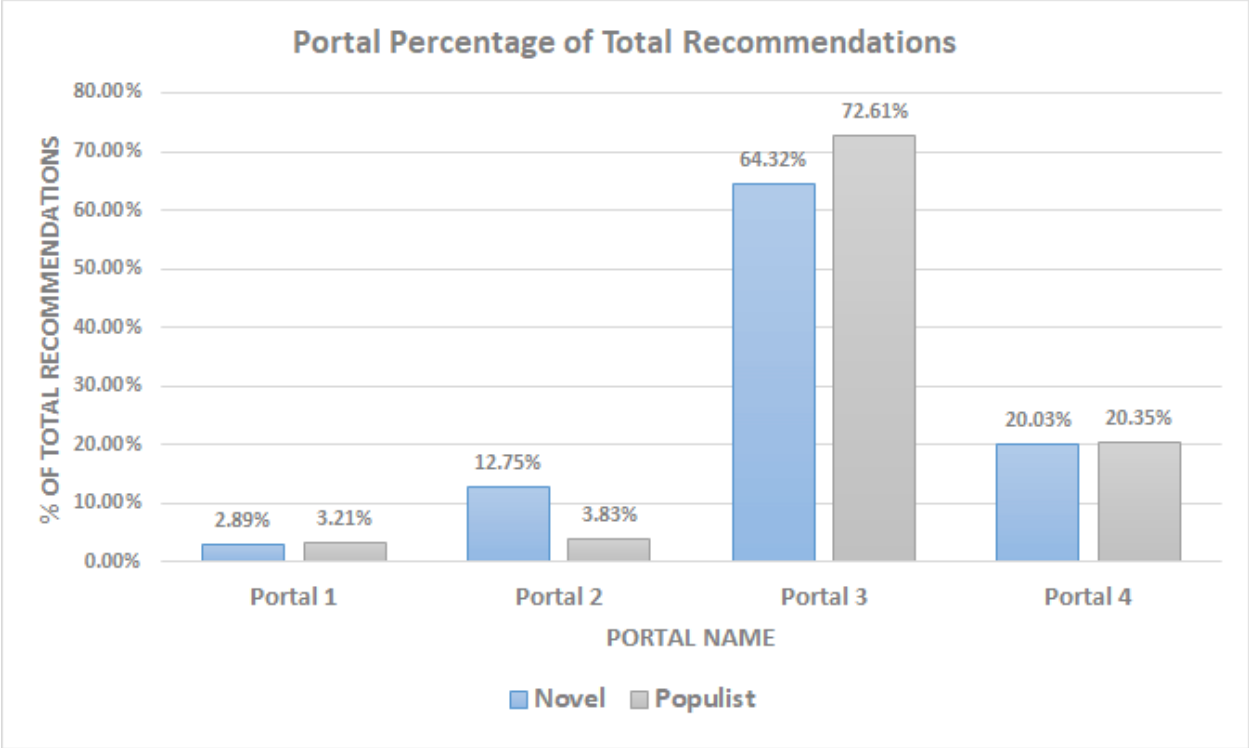


Figure 3.7: Breakdown of each portal’s total volume contribution to the total recommendations given as seen in the *combined* recommendations.

3.6.1 Candidate and Presented Recommendations

Due to the capture of the entire set of candidate recommendations, we can examine the effect of the distinct filters on the overall recommendation simulation. The results of the combined set, shown in Figure 3.6, demonstrate the impact that user behavior has on each filter, including the differences between each method (Novel and Populist).

For the Novel recommendations, we see that almost all (88.65%) of the recommendations that were extracted were not considered due to repeated recommendation for the session. This demonstrates the significant clustering of lesser used links between the vertices in the modeled user behavior, and a repetition of those values across the set of examined sessions. Not surprisingly, less than 1% (0.6% exactly) of the Novel recommendations were rejected due to the action already having taken place in the session. Overall, only 10.75% of the total recommendations considered during the simulation, and as mentioned above,

only 0.67% of those were found at all in the totality of the operations of the sessions examined. We feel that this demonstrates that a novel recommendation scheme, if followed (even in a small way), by the user would result in gains to the exploration coverage of the system, especially given the extremely small number of recommendations that were taken.

Within the Populist recommendations, we find a result nearly inverted from the Novel recommendation scenario, with a little over 77% of the rejected recommendations being due to the operation having already been done in the session. Since the recommendations were sourced from high weight links in the chain, this is to be expected. An interesting point is that there were no recommendations rejected due to being already recommended. This could be due to the sequence of checks in the rejection algorithm, wherein operations that were already done were checked for first. This order is the same for both Novel and Populist recommendations, so they are equally subject to that order of operation bias. All of this results in only 22.93% of the total candidate recommendations being eligible for presentation to the user, of which only 20.45% of those were found anywhere in the user’s set of session operations.

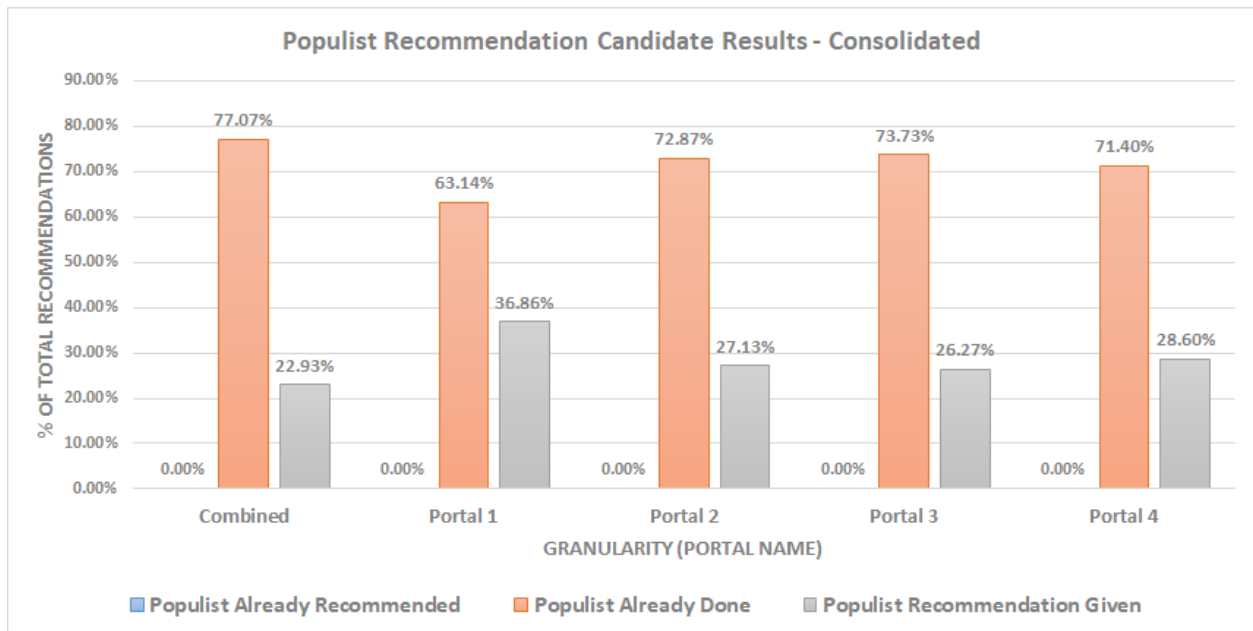


Figure 3.8: Populist recommendation candidate results.

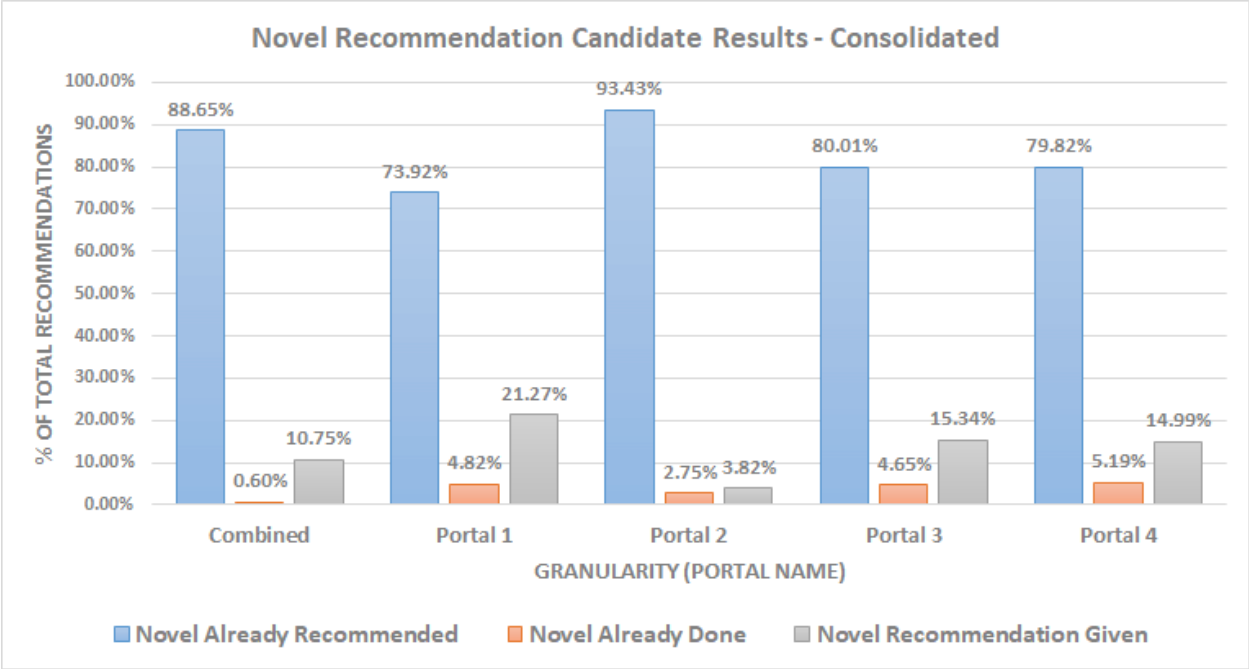


Figure 3.9: Novel recommendation candidate results.

As can be seen in Figures 3.9 and 3.8, the candidate results for the considered recommendations is very similar to the combined results within each of the portals. The pattern of rejection for Populist results being due to the recommendation being Already Done (for the Populist method), and Already Recommended (for the Novel method) remains consistent within the sub-portals in the system. The variability we see in each for each of the portals seems to be from the differing levels at which the portal activity matches the individual recommendation items, meaning that the recommendations (on a micro level) are more or less aligned with the overall system behavior.

From this data, we conclude that the pattern for recommendation rejection for candidate recommendations using the Populist method (Already Done) is mainly due to the incidence of popular items within the set of recommendations, while the rejection pattern for the Novel method is for rejection due to the frequent reoccurrence of lesser-visited items (Already Recommended).

3.6.2 Categories and Take Rates

The take rate values for the Novel and Populist recommendations represent our clearest metric of potential performance of each of these methods in improving the user’s overall knowledge of the system. In addition to the general take rates for the recommendations overall (for Novel and Populist), we also collected the take rates as broken down by the previously introduced categories.

The overall take rates for the combined dataset (seen in Figure 3.10, along with the portal specific take rates) are not unexpected when considering the method that the recommendations were created for each result. The Novel take rate of 0.67% is reasonable, given the intentional construction of the recommendations to favor typically unvisited areas of the system. Since, by definition of the method, users typically don’t visit these areas, they are subsequently not likely to have visited these items in their session. The Populist take rate of 20.45% is somewhat less obvious a result, especially given the method used to create the recommendations. Based on the recommendation of popular items, we might expect to see a higher percentage here. However, this does appear to represent that the sessions that were selected for recommendation analysis demonstrate specific item variation not adequately represented by the combined Markov chain, which is our original postulation. In other words, the specific items at each point that the users were suggested weren’t *exactly* what they ended up choosing. It is this lack of specific adherence that shows a potential benefit for the Populist method, given that its goal is to nudge the user toward a central tendency of behavior.

Investigating the portal specific take rates, we see no evidence of significant differences between the overall rates (in terms of general behavior) and the combined take rates. Overall, we consistently see a large disparity between the take rates for the Novel and Populist methods, as discussed in more detail above.

To establish an overall pattern of behavior for the take rates, we looked at the differences between the rate at which recommendations of a particular category were given ver-

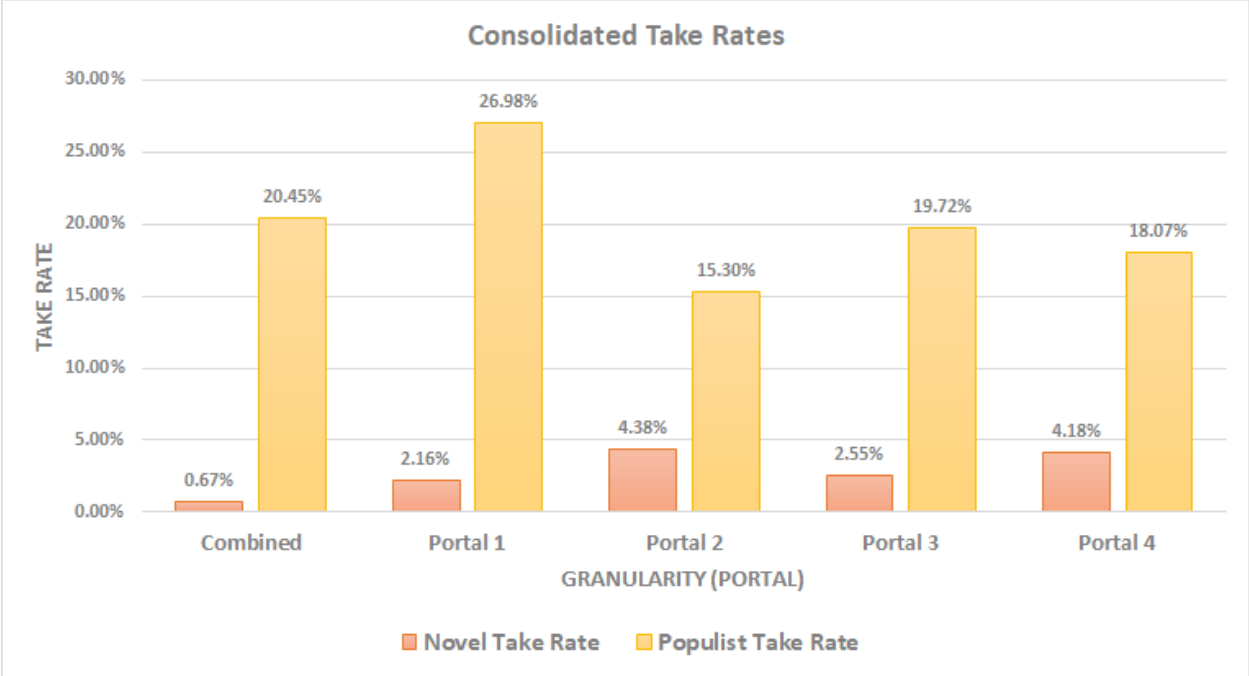


Figure 3.10: Take rates of Novel and Populist methods for each portal.

sus what percentage of each of the categories were taken. This is different than the overall take rate for the category, as this shows any patterns or disparities between the overall pattern of recommendations compared to the user’s preference for that category.

Table 3.5: Combined recommendation given/taken disparity.

Category	Populist Disparity	Novel Disparity
<i>Chart</i>	20.03%	4155.32%
<i>Data Source</i>	20.73%	435.00%
<i>Date</i>	-41.79%	-41.69%
<i>Filter</i>	-33.94%	-44.97%
<i>Hide Null Values</i>	-51.56%	331.82%
<i>Other</i>	2.13%	216.67%
<i>Variable</i>	5.56%	126.24%

Looking at Figure 3.12, which shows the Populist method category breakdown, we see small variations in the values, but no major deviations in the overall breakdown of the categorical recommendation. The disparity percentages between the given and taken recommendation categories can be seen in Table 3.5. Positive percentages indicate a higher inci-

dence of deviation from the recommended category rate (users chose from this category at a higher rate than was recommended) whereas negative values indicate the opposite.

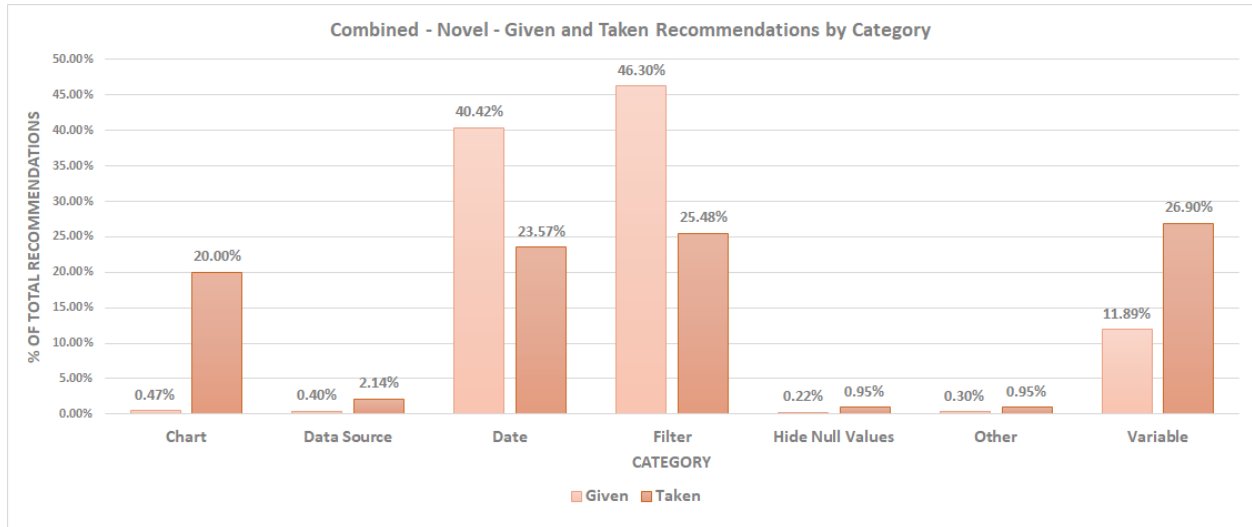


Figure 3.11: Combined - Novel - Given and Taken recommendation categories (*Chart*, *Data Source*, *Date*, *Filter*, *Hide Null Values*, *Other*, and *Variable*), listed as percentages of the total Given and Taken recommendations, respectively..

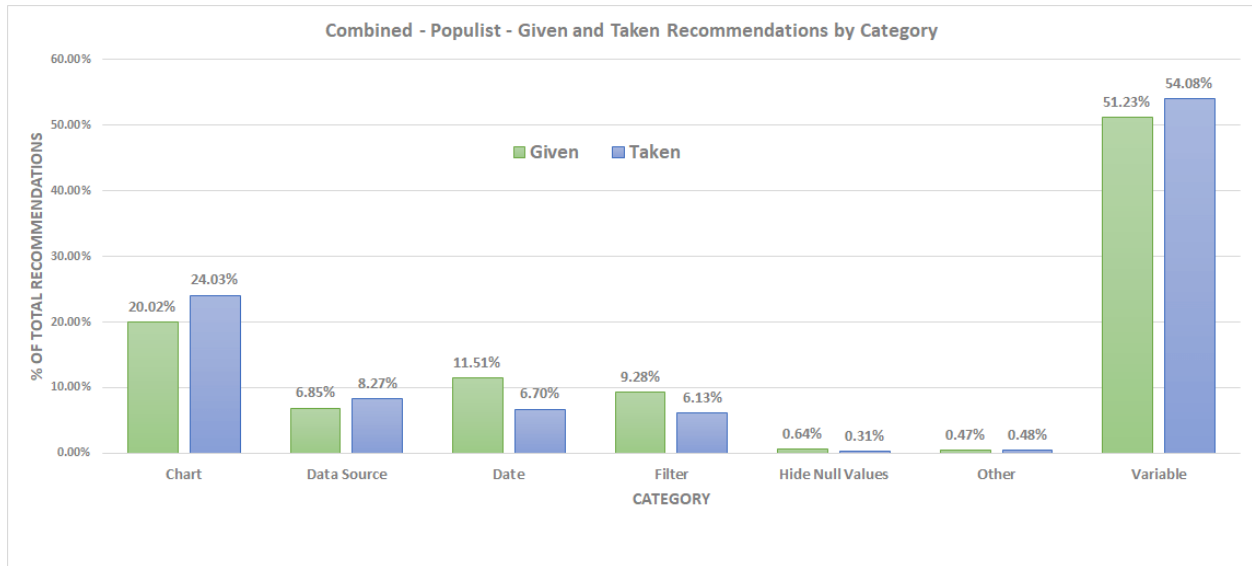


Figure 3.12: Combined - Populist - Distribution of Given and Taken recommendation categories (*Chart*, *Data Source*, *Date*, *Filter*, *Hide Null Values*, *Other*, and *Variable*), listed as percentages of the total Given and Taken recommendations, respectively.

The most obvious disparity is that between the given/taken rate within the *Chart* category for the Novel method, with the over 4,000% difference between the occurrence of

Chart values in the recommended set and the presence within set of operations done by the user. Additionally, we see 100%+ disparity values for *Hide Null Values*, *Other*, and *Variable* categories. Overall, these disparities demonstrate that the behavior of users, as measured by the operations content of their sessions, is much different than the categorical distribution of randomly selected values in the bottom 50% of the selection at each node. While we would expect to see some level of mismatch, given the randomness of the selections, the great disparities that we see are well in excess of what might be expected from simple distribution variance. Although it is difficult to determine the exact effect that the random choice set had on the outcome, we acknowledge that the greater disparities are possibly due in part to that choice.

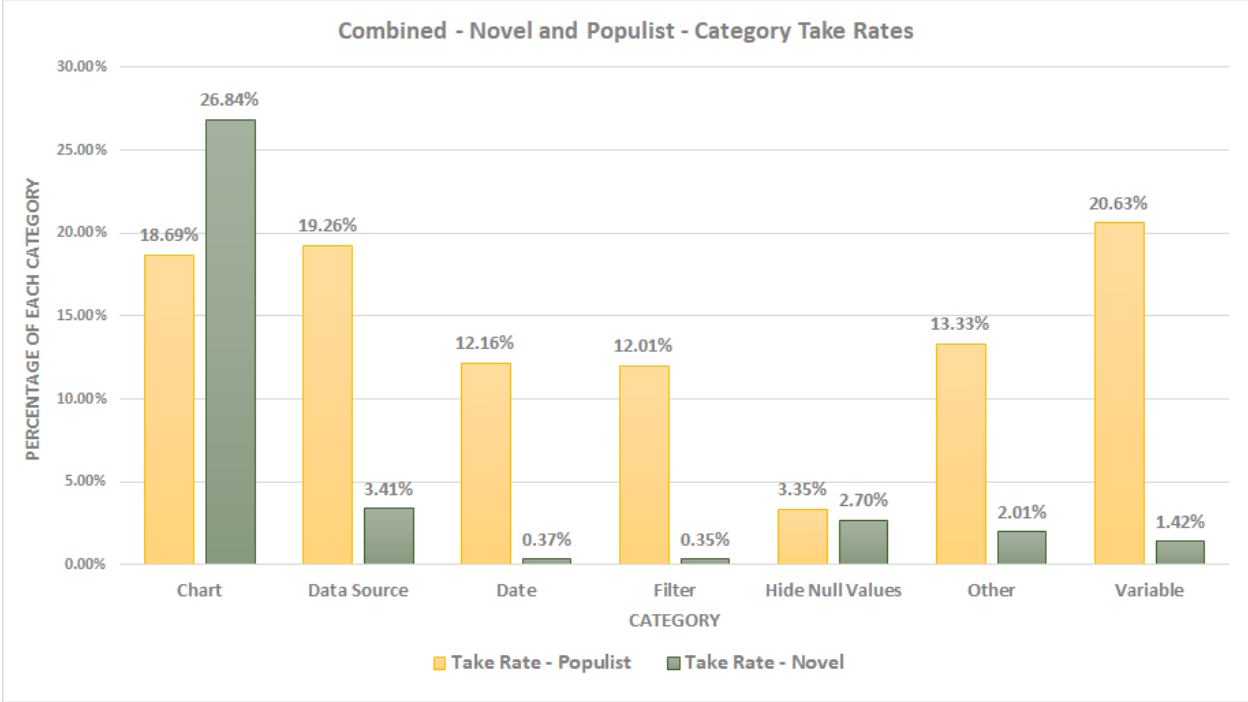


Figure 3.13: Combined take rate distribution by category and Novel and Populist methods.

The take rates were also extracted for each of the individual portals to determine if there was evidence of any interesting variability. Note that, as stated previously, the portal representation in the data is not uniform across the entire sample, as the subdivided data was extracted from the randomly chosen 25% sample of the full set of user operations, not

individually by portal. For reference, this breakdown can be seen in Figure 3.7. The Novel take rates, divided by portal (including the combined rates) can be seen in Figure 3.14, while the Populist rates can be seen in Figure 3.15.

Although the rest of the metrics in the analysis have been relatively consistent, it is within the portal specific categorical take rates where some variability emerges. Note that the portal samples were taken from the full set of values, so their ratios in the breakdown are the same as is shown in 3.7. This is important to consider when using these results when attempting to use them for non-internal conclusions.

Overall, the variability within each of the categories is of a sufficient amount to at least say that the behaviors within each of the portals is different. There are several potential reasons for these internal differences, including non-overlapping variable data (meaning that certain values only exist in one portal and not another), different data sources (which also influences the variable data), and the obvious fact that the user groups of these systems are mutually exclusive. Based on the subset values that we procure from excise selection of these values, we feel that a larger sample size, which would come from an expanded data collection effort, would be required to conduct a full analysis of the data. This is primarily due to the overwhelming occurrence of data for Portal 3, which stems from the fact that this portal was in use for much longer than the other portals, and as such, had more (and more mature) users as part of its user group.

3.7 Conclusions and Future Work

3.7.1 Novel Method and *Already Recommended*

Although we designed both the Novel and Populist recommendation methods to avoid bombardment of the user with the same items over and over, the extremely low take rate for the novel recommendations, as well as the high rejection to already recommended values, leads us to the conclusion that reiterating duplicate recommendations for the Novel method may not be contrary to the goal of user education. We would still retain the *Al-*

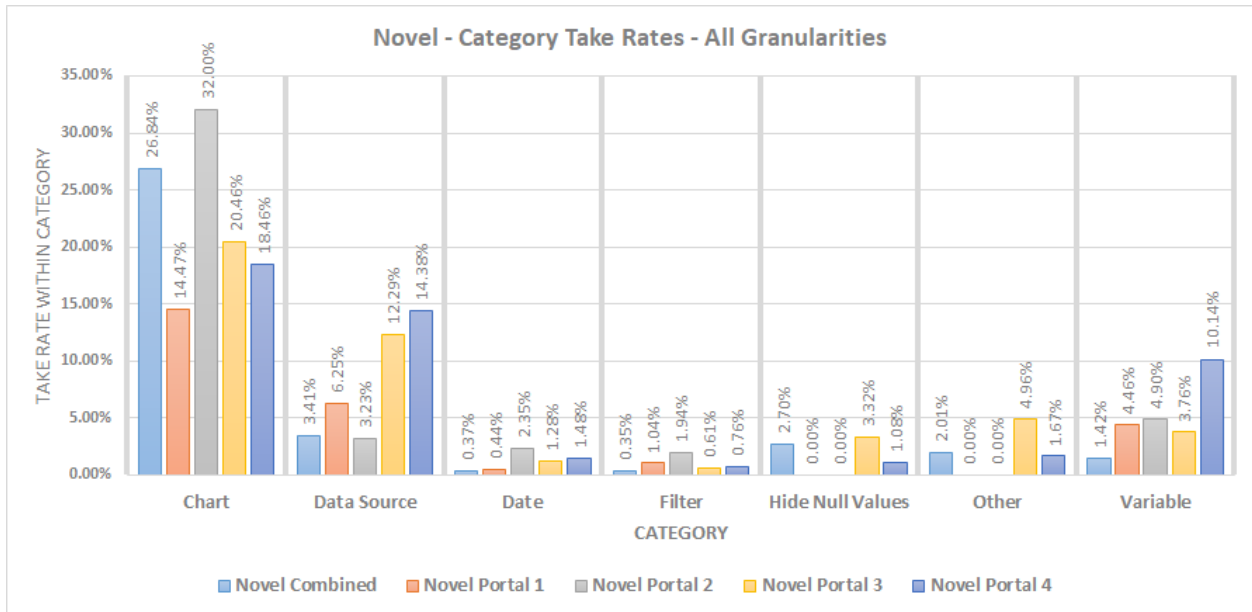


Figure 3.14: Novel categorical take rates, listed by portal.

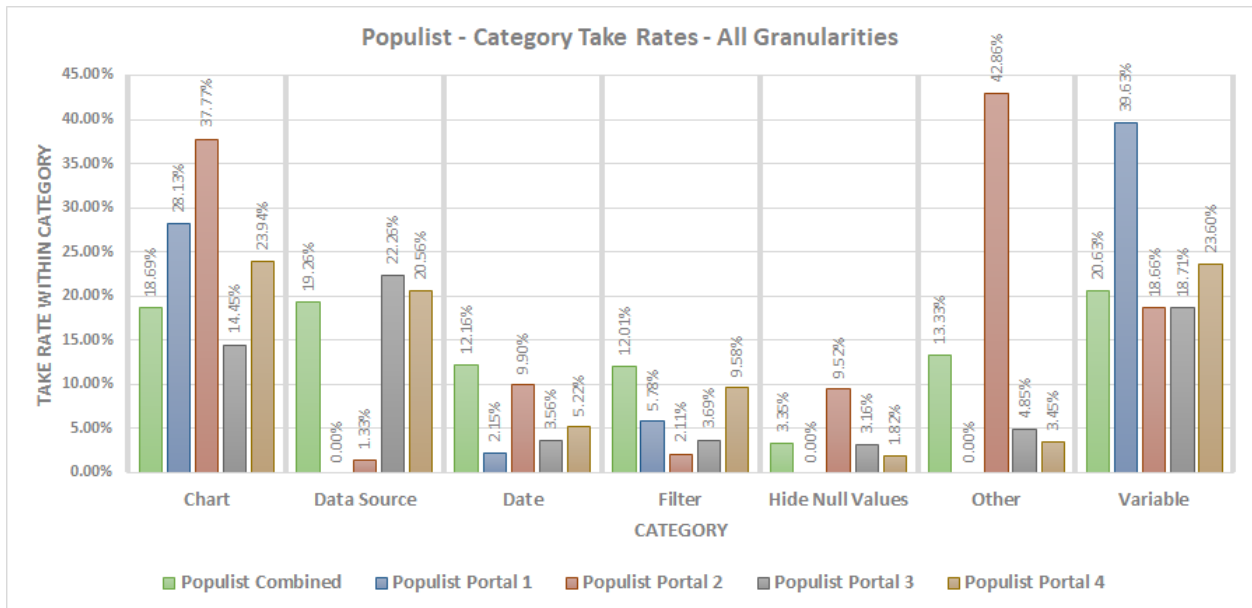


Figure 3.15: Populist categorical take rates, listed by portal.

ready Done restriction for recommendations, which would at least prevent annoying the user with things that they'd already chosen. While not definite, we believe that this modification to the algorithm would result in a higher overall take rate, if for no other reason that we would repeatedly recommend items which were novel within the system.

3.7.2 Potential Practical Benefits

In addition to the specific patterns and slight changes that might result from modifying the method for eliminating duplicate recommendations from the session (as mentioned above), we find that given the individual variability of the specific items which we are recommending, there is a great potential benefit in promoting system familiarity by implementing this system in a real user environment. The current technical limitations of our post-event analysis makes full prediction of the benefits difficult to quantify, but we base this idea on the wide array of possible values within the set of recommendations combined with the low take rates witnessed in both the Populist and Novel sessions.

Detection of the user's ability level (possibly using a form of Item Response Theory (IRT)[4] that uses previous behavior as a baseline), and an automatic selection of a Novel, Populist, or hybrid navigation suggestion set based on this result could prevent giving less relevant or useful results to a user.

3.7.3 Future Work

Based on our findings, we plan to explore further the possibility of implementing the two recommendation methods as 'wizard-type' educational tools for both new and existing users. The Populist method, with its tendency to drive users to popular items within the system, seems well suited to new users who do not have a full conceptual picture of the available data points and might benefit from a guide of this kind. For the Novel method, targeting of longer-term users would, if time allowed, give a user-specific, but generally focused tour of the lesser examined items in the system, possibly exposing unknown or at

least unusual navigation and analysis patterns for existing datasets. Ultimately, a system in which the users are aware of the general content and behavior of the data leads to more intentional analysis and may help prevent unintended meandering or unintentional ignorance of potentially useful or interesting data points.

In our work, we focused solely on single ‘in the moment’ recommendations that were reevaluated at each point in the user’s session. However, we recognize that given the data available to us, construction of a full ‘guided tour’ (or playlist of recommendations [2, 8]) of the data within either the Novel or Populist methodologies is possible and could provide benefit for the user. This process would involve a much more passive user that would be willing to navigate a pre-fabricated path instead of a set of operations that they chose themselves. It would be interesting to determine and test the metrics of coverage in the system, as well as how to create these tours with different goals in mind. The criteria for creating these sets could be based on several factors, and provide a more localized (and possibly standard and repeatable) introduction (or reintroduction) to the dataset. Items that ‘make sense’ within the user’s current examination pattern (measured by typicality[3], possibly) may also aid in reducing the occurrence of novel items that are outside of a bounded list of what is reasonable.

In addition to the standard ranking methods for choosing Novel and Populist values for recommendation, a more complete ontological view of the data[19], with consideration of the context of not only the local graph network, but also of the concept transitions (e.g. vehicle-related data to person-related data) would likely provide a more nuanced and personalized recommendation scheme. However, this may lessen the overall averaging effect that the our Novel and Populist methods seek to have, though this may not always be an undesirable outcome, depending on the dataset familiarity of the individual user.

References

- [1] ANDERSON, C. R., DOMINGOS, P., AND WELD, D. S. Relational Markov models and their application to adaptive web navigation. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02* (2002), 143.
- [2] BACCIGALUPO, C., AND PLAZA, E. Case-based sequential ordering of songs for playlist recommendation. *Advances in Case-Based Reasoning* (2006), 286–300.
- [3] CAI, Y., LEUNG, H.-F., LI, Q., TANG, J., AND LI, J. Recommendation based on object typicality. In *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM '10* (New York, New York, USA, 2010), ACM Press, p. 1529.
- [4] CHEN, C. M., LEE, H. M., AND CHEN, Y. H. Personalized e-learning system using Item Response Theory. *Computers and Education* 44, 3 (2005), 237–255.
- [5] CHO, Y. H., AND KIM, J. K. Application of Web usage mining and product taxonomy to collaborative recommendations in e-commerce. *Expert Systems with Applications* 26, 2 (feb 2004), 233–246.
- [6] DUAN, L., STREET, W. N., AND XU, E. Healthcare information systems: data mining methods in the creation of a clinical recommender system. *Enterprise Information Systems* 5, 2 (may 2011), 37–41.
- [7] GARCIN, F., DIMITRAKAKIS, C., AND FALTINGS, B. Personalized news recommendation with context trees. *Proceedings of the 7th ACM conference on Recommender systems - RecSys '13* (mar 2013), 105–112.
- [8] GERMAIN, A., AND CHAKARESKI, J. Spotify Me: Facebook-assisted automatic playlist generation. *2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)* (sep 2013), 025–028.
- [9] HARIRI, N., MOBASHER, B., AND BURKE, R. Query-driven context aware recommendation. *Proceedings of the 7th ACM conference on Recommender systems - RecSys '13* (2013), 9–16.
- [10] KHRIBI, M. K., JEMNI, M., AND NASRAOUI, O. Automatic Recommendations for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval. *2008 Eighth IEEE International Conference on Advanced Learning Technologies* (2008), 241–245.
- [11] MANOUSELIS, N., DRACHSLER, H., VUORIKARI, R., HUMMEL, H., AND KOPER, R. Recommender systems in technology enhanced learning. In *Recommender systems handbook*. Springer, 2011, pp. 387–415.
- [12] MAYEKU, B. Enhancing Personalization and Learner Engagement through Context-aware Recommendation in TEL. 413–415.

- [13] MOBASHER, B., COOLEY, R., AND SRIVASTAVA, J. Automatic personalization based on Web usage mining. *Communications of the ACM* 43, 8 (2000).
- [14] RAY, W. 5. probability and random processes: Problems and solutions. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 156, 3 (1993), 504–504.
- [15] SCHAFER, J. B., KONSTAN, J., AND RIEDI, J. Recommender systems in e-commerce. *Proceedings of the 1st ACM conference on Electronic commerce - EC '99* (1999), 158–166.
- [16] SMITH, R. K., GRAETTINGER, A. J., KEITH, K., AND PARRISH, A. Identifying High Frequency Crash Locations: Empowering End-Users with GIS Capabilities. *ITE Journal* 77, 1 (2007), 22–27.
- [17] STECK, H. Item popularity and recommendation accuracy. *Proceedings of the fifth ACM conference on Recommender systems - RecSys '11* (2011), 125.
- [18] TANG, J., GAO, H., HU, X., AND LIU, H. Context-aware review helpfulness rating prediction. *Proceedings of the 7th ACM Conference on Recommender Systems* (2013), 1–8.
- [19] TARUS, J. K., NIU, Z., AND MUSTAFA, G. Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning. *Artificial Intelligence Review* 50, 1 (2018), 21–48.
- [20] VERBERT, K., AND DRACHSLER, H. Dataset-driven research for improving recommender systems for learning. *Proceedings of the 1st International Conference on Learning Analytics and Knowledge* (2011), 44–53.
- [21] WILSON, D. C., AND SEMINARIO, C. E. When Power Users Attack : Assessing Impacts in Collaborative Recommender Systems. 427–430.
- [22] ZHU, J., HONG, J., AND HUGHES, J. G. Using Markov models for web site link prediction. *Proceedings of the thirteenth ACM conference on Hypertext and hypermedia - HYPERTEXT '02* (2002), 169.

CHAPTER 4

ARTICLE 3 - TOUR GUIDE: PROVIDING WORKFLOW OPTIMIZED RECOMMENDATIONS TO USERS OF AN ANALYTICS SYSTEM

4.1 Introduction

There are over 2,500 active users of the web-based analytics system commonly referred to as ADVANCE (Advanced Dashboard for Visualization Analysis and Coordinated Enforcement) [17]. This system is based on the CARE analytics engine [13] that uses a custom process for extracting, translating, and loading data into a format that allows for extremely fast analysis of relationally stored data. This system is used by myriad types of users, including law enforcement officers, epidemiologists, roadway engineers, government personnel, and many others. Data analyzed and provided by the system includes ambulance run reports, vehicle crash reports, and citations.

One of the primary use cases for this system is to allow for approachable analysis of complex data by a relative layperson in statistics. More specifically, this system is used for data-based decision making which often includes choosing potential countermeasures to implement to reduce negative effects (such as motor vehicle crashes)[18]. As the volume and complexity of collected data increases, tools for examining and distilling the lessons presented by this data are extremely important. Intuition and “gut feelings” that are based on years of experience in a field, dealing with the activities that create these datasets, are

deceptive and often inaccurate or misleading. ADVANCE (and its variants) provides a mechanism for extremely fast and accurate analysis of data.

Leveraging analytics system usage data and decision-tree classification techniques, we constructed a recommender system called Tree-Optimized User Recommendation (TOUR) Guide, which uses the binary classification decisions that led to a discovered preference item, export of data from the system, as a basis for constructing and recommending items to present to the user for consideration.

4.2 Research Background

Throughout the study of recommender systems, researchers have improved the efficiency and overall experience of using high item volume systems by presenting users with the most relevant data to their needs. Much of this effort has been focused on a small, ever improving set of available datasets, such as those provided by Netflix [2] or MovieLens [9]. The research on these datasets has yielded a valuable set of criteria for selecting appropriate recommender systems and applying them in the most effective manner. As the sophistication of the knowledge for recommending items improves, this abundance of knowledge bears application to other datasets. However, each dataset and user group presents unique challenges.

Many times, the size and complexity of a dataset has served as the motivation for the development or application of a recommender system. Analytics datasets, though they provide the potential for insights into the subjects that are their focus, are sometimes not fully utilized due to their size or complexity. It is often the case that painstakingly collected data goes unused or underutilized because of either its presentation complexity or simple data volume. Even advanced users, who have extensive knowledge of the domain and the data contained therein, could benefit from the body of knowledge offered by the various recommender system techniques.

Recommender systems have traditionally been based on user to user (collaborative filtering) or item to item (content-based filtering) where either the user or the items they've selected are compared to other users or items to determine relevant recommendations. There are hybrid approaches that will use combinations of these (and other) methods to provide recommendations. Recently, there has been an increased focus on the use of machine learning and artificial intelligence, including the use of decision trees in various capacities[15, 19]. These also include ontology-based trees[3] and addressing the *user cold start problem*[7].

Providing users sequential recommendations (where order *and* content matter)[6] is an important consideration, especially when the order of the events can have such an impact on the outcome. Within budget-conscious government agencies, the cost of organizational instruction on the datasets, especially when that cost is not negligible and the user is inexperienced (i.e. hasn't contributed preferences to the system that might be used as source data) presents issues when considering how best to direct users to a desired outcome[20] that may be addressed with a more data-driven approach to fast results.

4.2.1 System Usage Extraction

As the first step to construct TOUR Guide, we created and deployed a parallel logging system to capture the system usage. Coincident with the migration of the ADVANCE-backend (the CARE engine) from major version 9 to major version 10, this diagnostic logging process was inserted to capture activity sent to the CARE web services. The logging was captured in two database tables, described below:

tblOperationLog (User Operation and Context)

- id (integer)
- datetimeOccured (datetime)
- username (varchar(50), obfuscated (hashed))

- `userOperation` (`varchar(75)`) – CARE web service method that was called
- `portalName` (`varchar(50)`) – Website that was using the web service (this allows identification as to whether it was ADVANCE or one of the other variant portals)
- `logSessionId` (`varchar(50)`) – This is a Globally Unique ID (GUID) that identifies the user’s session. Prior to the implementation of this explicit field, a process to identify sessions was manually implemented in code that identified user activities as being part of the same session if the duration between each activity did not exceed the user login timeout. This is a fairly reliable method, since if the user had no logged activity, they would have been automatically logged out and a new session would have been detected because of the break in activity.

tblOperationLogParameters (User Operation Parameters)

- `id` (integer)
- `OperationLogId` (integer, foreign key)
- `parameterName` (`varchar (50)`)
- `parameterValue` (`varchar(100)`)

The source dataset was collected as part of debug and telemetry for the system over approximately thirty months. There are 13,951,227 records in the *tblOperationLog* table and 18,675,936 records in the *tblOperationLogParameters* table. The combination of the relationship of these records makes up the entirety of the general user behavior in the ADVANCE system.

There are two primary categories of this collected data: *dataset filters* and *dataset display preferences*. Dataset filters refers to parameters such as start and end date, predefined filters (built manually by the creators of the datasets), and one or more filtering criteria (chosen from the available categorical variables of the dataset). These are shown in the

left pane of Figure 4.2. The dataset display preferences, shown in the four large panels, are composed of the four available variable display tiles and the different display methods (graph types) that can be used to visualize that data. Once the user has used the filters to pare down the dataset, he can then explore the frequencies of up to four unique variables in that set.

User permission and variable visibility is an important factor in the system. Specifically, in ADVANCE there are several state and local agencies that have either their own public safety dataset (e.g. vehicle crashes) or a selective view into the larger dataset based on their location or organization membership. For example, if the user is a member of Agency A, then members of Agency B cannot see data specific to that user’s area, and vice versa. In most cases, this is either a data ownership decision or an explicit request from the agencies involved. These built-in assumptions regarding who can see what enable us to more easily account for small variations in the way each portal is implemented and differences between what variables might be available.

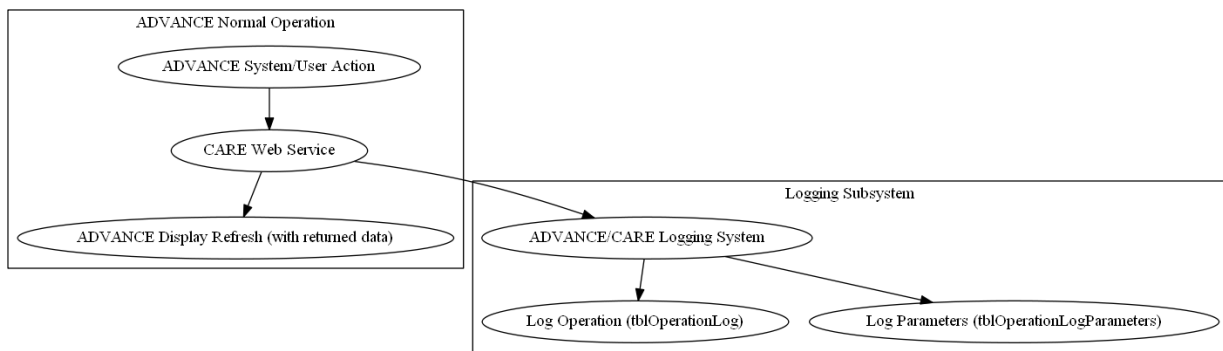


Figure 4.1: Diagram of the logging process for ADVANCE.

The diagram, shown in Figure 4.1 shows the process for logging the activity in ADVANCE. The CARE web service calls are sent to the logging subsystem, which logs the method called as well as all of the key/value parameters.

The primary variables of interest, that are easily discernible as inputs from the main page of ADVANCE:

- Start Date
- End Date
- Filter
- Filter Variable
- Charts 1-4 Variable
- Charts 1-4 Type (bar, pie, etc.)

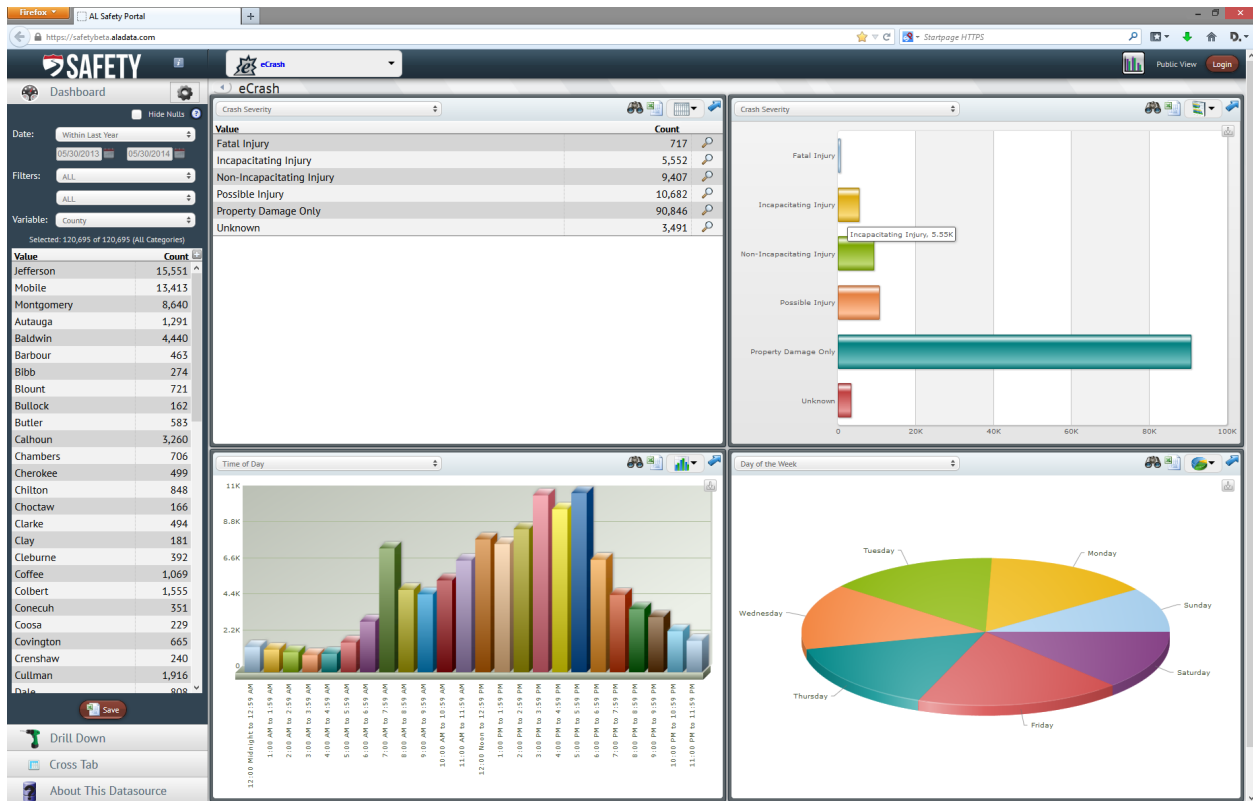


Figure 4.2: Screenshot of *ADVANCE*. The *dataset filters* are displayed in the left column and the *dataset display preferences* are displayed in the four large panels to the right.

4.2.2 System Usage Analysis and Motivation

Users of the *ADVANCE* system have an immense amount of distinct analysis outcomes that they can yield by navigating through the combinations of filters, variables, and dis-

play options (e.g. charts). Throughout the time that the user is navigating around the system (their *session*), the user is (ideally) making choices that reflect their dataset research goals, if they have any for that particular session (or portion of the session). While user intent is not immediately apparent at the point of each choice, users are able to export any of the results that they find interesting or useful to Microsoft Excel or a comma-separated value (CSV) text file for further analysis. This process follows a standard workflow for users of this system of the following:

1. Formulate or be given an analysis question to answer from the dataset.
2. Configure the analytics environment in a way to find the answer.
3. Record separately (copy and paste) or directly export the data for inclusion into a report or for further separate analysis.

This export is the only explicit user action that could be used to denote preference for a particular composite of choices, and is especially important in the context of the overall workflow within the particular domain's individual portal[21]. However, simply looking at the state of the system at the time of export does not reflect the sum of the data awareness of the user at the time, since the user may have received benefit or education from the lead-up choices that led to this point. However, it remains that the sole explicit choice of preference (by retaining the data after the session ends) is the export operation. With this in mind, it is important to understand the placement of this export in the context of the session to determine if there is any discernible pattern. The Figures 4.3, 4.4, and 4.5 show the aggregate export occurrence, listed by the percentage at each point of the session of total export operations. Note that events listed at point 1 represent exports that occurred between 95% and the final event of the session.

Examining the pattern of export values, we see that the frequency of export events increase as the position in the session increases. This is especially true with the single export sessions, with over 52% of the exports occurring in the last 20% of the session. These

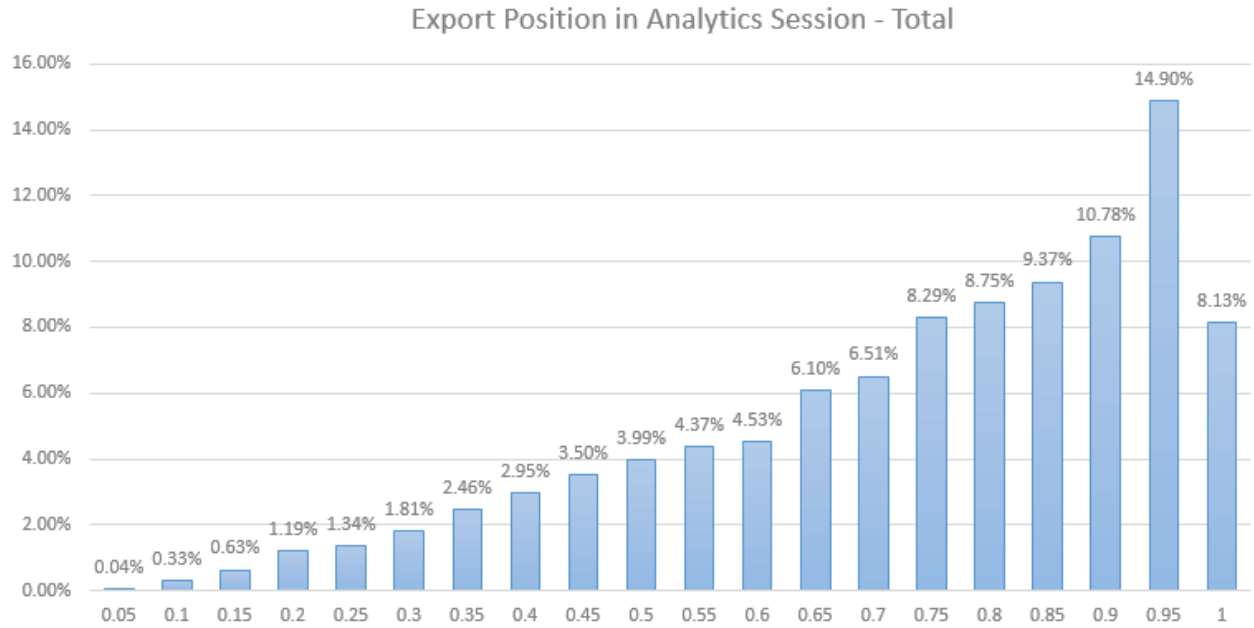


Figure 4.3: Session export position for all sessions (single and multiple).

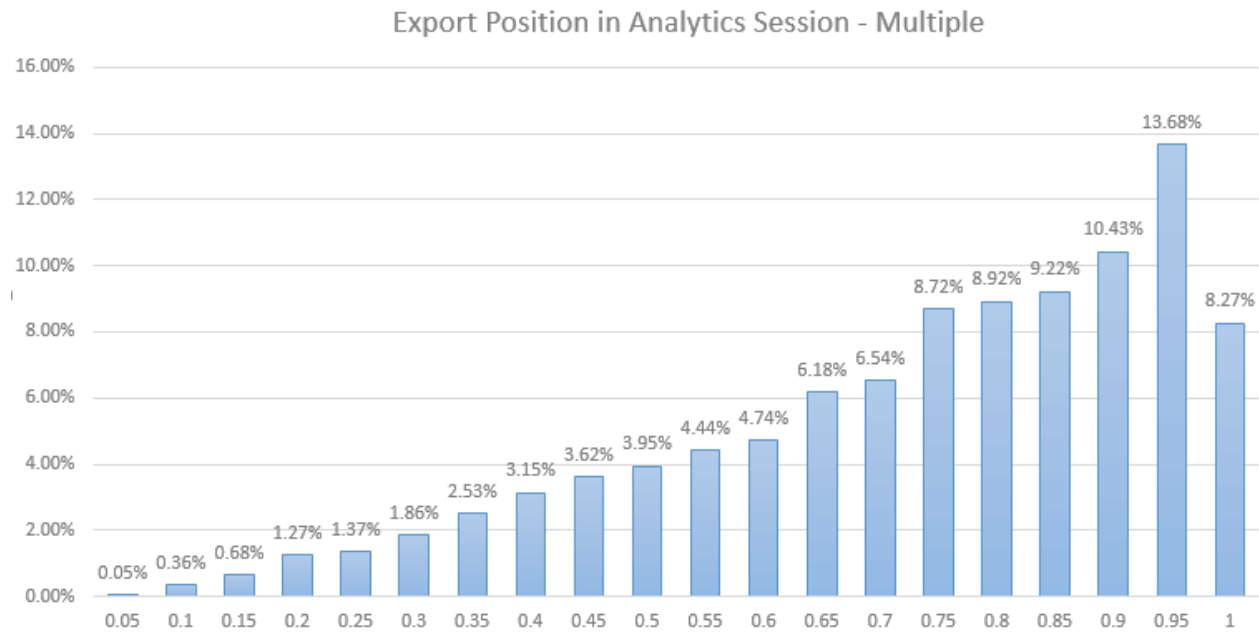


Figure 4.4: Session export position for sessions with multiple exports.

frequencies are calculated as percentages of all sessions, so this pattern is consistent for sessions of any length. This data suggests that users correlate export of the results of their session with a successful session (in accordance with their goals), as they do not continue to use the system long after they've exported. Even in the case of multiple exports per

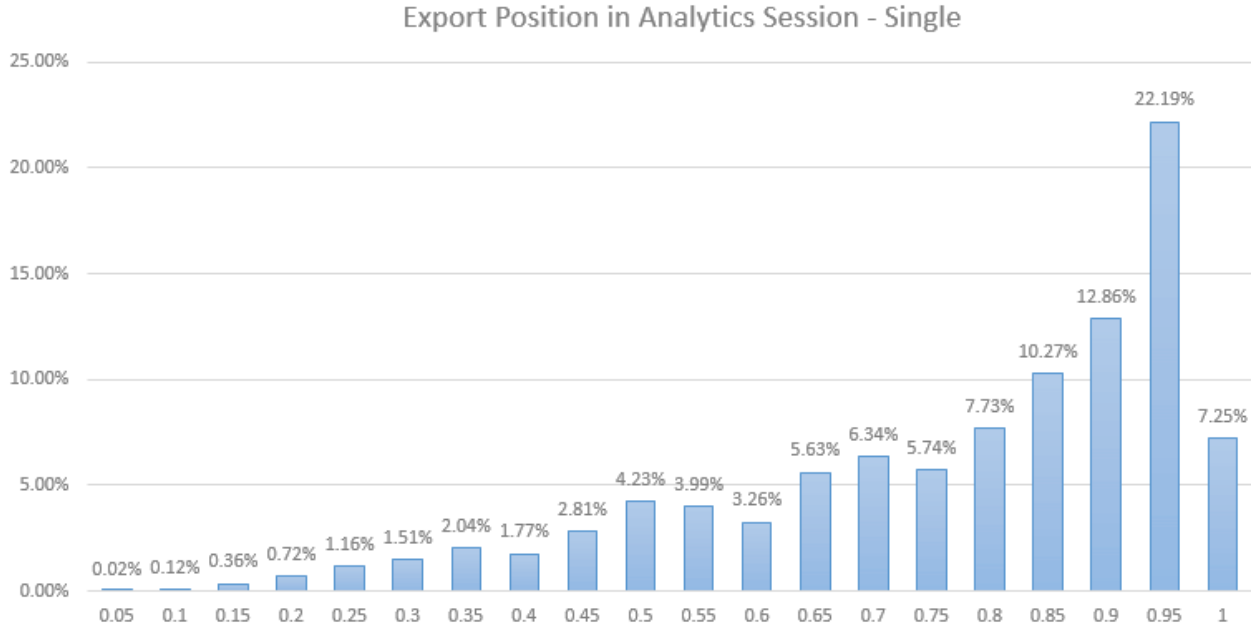


Figure 4.5: Session export position for sessions with one export only.

session, export frequency increases along with the timeline to the termination of the user’s session. With 86.6% of the user sessions including multiple exports per session, this indicates that even multiple exports still occur at the terminus of the session.

With this support for exporting of data as a preference metric in mind, exploring methods by which to direct the user to results that include or terminate in an export could lead to a more efficient user experience while still giving the user the same analytical value. To this end, we examined the decisions that led users to sessions of this type through the use of a decision tree which held a binary classification of a session that either *did* or *did not* end in an export.

4.3 TOUR Guide

Taking some inspiration from a traditional travel agent or tour guide, including previous work related to automation of this analogous process[11], we constructed a recommender system based on creation and navigation of a decision tree. The primary goal of this system is to provide recommendations to users that are both relevant to their current

place in their session and that optimizes for a set of decisions that lead to an export to Excel/CSV. This system, called Tree-Optimized User Recommendations (TOUR) Guide, is based on the classification decision on whether the session ended in an export and constructs a multitude of localized decision paths that are then presented to the user. These recommendations are recalculated at each point in the user’s session based on the delta between the current session state and the previous session state. Construction of this system involves several steps, including extraction of the features from the user session data, building the decision tree, and constructing the traversal and recommendation subsystem which provides the recommendations.

4.3.1 Feature Extraction

Beginning with the identification of the universally present values in the system, we were able to distill the values that were both consistently available and were extractable into either continuous or could be indicated with a binary preference marker. Using the original log data, we wrote a secondary extraction process which iterated over the values and assigned their presence into the available columnar categories. Overall, we extracted the major items related to the analysis process, including the operation conducted (which translates to the web service call), variables, portal (different users can originate from different web sites based on their agency affiliation), and data source (the source dataset that was used to build the analyzed variables). The complete listing of the extracted features, with descriptions, are shown in Table 4.1.

The *exportedToExcel* variable was removed in order to pass to the classifier module as the classification parameter used for splitting. As such, this is categorized as a binary classification, since we are only concerned whether features contribute to the choice to export or not. Additionally, this removes the possibility of the model to be able to simply optimize to the values that have the classification marker as true. Using the Python scikit-

Table 4.1: Description of extracted features.

Name (Variable Name)	Description
Weekday (<i>weekday</i>)	Integer value 1 to 7
Hour of Day (<i>hourofday</i>)	Integer value 1 to 24
User Operation (<i>op_{OPERATIONNAME}</i>) (6 total)	Binary indicator of whether operation named at the column was called in this row
Portal Name (<i>portal_{PORTALNAME}</i>) (58 total)	Binary indicator of whether portal named at the column was the one visited for this method call
Age (<i>age</i>)	Difference between date of analysis and start date of dataset filter
Duration (<i>duration</i>)	Difference between end date of dataset filter and start date
Exported To Excel (<i>exportedToExcel</i>)	Binary indicator whether session resulted in call to export data to Excel file
Variable (<i>variable_{n}</i>) (1 to 718)	Binary indicator of whether variable at that index was parameter in method call
Data Source (<i>datasource_{n}</i>) (1 to 44)	Binary indicator of whether data source at that index was the parameter in method call

learn library (specifically the *DecisionTreeClassifier* module)[14], the training data was then used to inform the model in order to allow it to classify the test set.

Given the above total available values, there are 814 total analyzed features. *Variable* and *Data Source* make up the bulk (754 of the 814) of the features. These features are binary classification features that indicate presence of the selected row in the feature’s class. The remaining features are non-binary, and are ordered along a logical spectrum (e.g. days of a the week). In addition to the previously mentioned performance reasoning, the significant presence of binary classifiers was also a motivator.

Several of the data rows were excluded, primarily to remove either automated (non-human-initiated) or non-relevant (e.g. logging in and out, etc.) web service calls. The majority of the recorded values (approximately 93%) originated from calls necessary to populate a GPS locator service, which populated real-time positions on a map. These were non-interactive calls to the web service, and thus do not contribute any user preference in-

formation. The resulting query also only included only the rows which contained values for both the Data Source and Variable. These were the most numerous and were the best available representative values that were user driven, as well as providing a set of consistent inputs for analysis as this eliminates the need to account for missing variables (which would require inserting “dummy variables” to preserve consistent dimensionality).

4.3.2 Decision Tree Construction and Export

This decision tree is built from a standard Classification and Regression Tree (CART) [4], with the configured metric for the process of assigning the node values being the *Gini Impurity*, calculated as:

$$Gini = 1 - \sum_i p_i^2 \quad (4.1)$$

Gini impurity measures how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset. The Gini impurity can be computed by summing the probability of an item with label i being chosen times the probability of a mistake in categorizing that specific item. The impurity reaches zero (its minimum) when all cases in the node fall into a single target category.

Information Gain Entropy is also sometimes used in CART-based decision tree analysis. Gini impurity was chosen, in this case, mainly due to the reduction of computational complexity and execution time cost. Gini impurity has been found to have little practical difference in analytical results (in less than 2% of cases)[16] and was deemed appropriate given the increased number of iterative analysis runs that its use permitted.

Using the DecisionTreeClassifier module classification tree, the result was exported to a GraphViz *dot* file for visual analysis and initial process validation. A subset of the graph, shown in Figure 4.6, is an export that was capped at 7 levels of the tree. Items below the threshold of that export are shown with gray boxes that contain ellipses. The automated export system that was constructed as part of this analysis produces an export

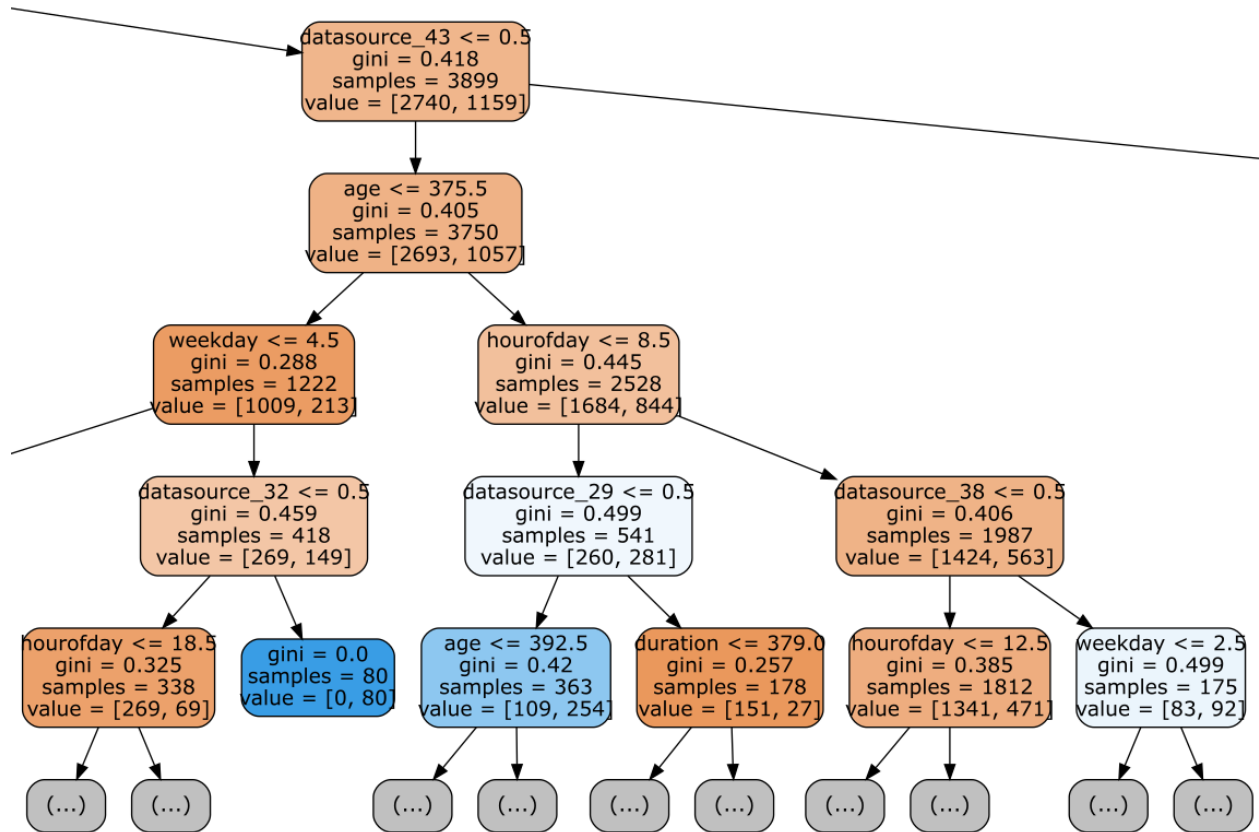


Figure 4.6: Subset of a full tree (with export specificity capped at 7 levels).

at the specified number of levels deep based on the values in the configuration file. This was especially helpful for troubleshooting export errors without having to navigate the full tree, which contains tens of thousands of nodes, and is difficult to navigate swiftly.

Samples and *value* are the number of samples considered in the decision for this node and the split of the values into each decision category, respectively. In order to allow easier adjustment of the primary parameters, and to promote simulation reproducibility[10], a simple YAML-based configuration file, which is split into five sections: query, extract, tree, export, and simulation. Comments are included in the file for easier adjustment, as well. The entire file can be seen in Listing 4.1. A copy of the simulation file that is used for each simulation is also copied to the log output directory for that simulation run to allow for repetition of that specific configuration if necessary.

Listing 4.1: TOUR Guide Configuration File

```

query:
  debug_echo: false
#options for data export and split
extract:
  test_size: .25
  train_size: .75
  extract_simulation_set: true
  simulation_extract_ratio: 0.25
#options for decision tree
tree:
  #options (gini, entropy)
  criterion: gini
  min_samples_leaf: 1
#options for export, particularly of the DOT/GraphViz tree
export:
  #depth of tree to create explicit files for.
  #0 will result in no intermittent tree exports
  depth: 10
  #options (true, false)
  full_export: true
simulation:
  #limit of sessions to run simulation over. 0 means no limit.
  clear_recs_given: false
  limit: 0
  recommendation_threshold: 0

```

Along with the configuration file, which is stored in a log folder that is identified by the Unix time at the start point of the simulation run, several other files are present. Included in this folder are a general log file (that records the shape of the arrays that make up the input data, the prediction accuracy, and the MCC), a configurable number of output GraphViz files that represent different levels the binary decision tree, a full GraphViz

representation of the tree, a textual representation of the binary tree, a summary set of aggregate values that shows the bounded values for *weekday* and *hour of day* at each of the leaves, and a file that records the output of the simulation run. Most of these files are either translated from the in-memory representation of the graph or are output from some of the parallel support data structures that are constructed to set up the simulation.

Prediction Accuracy

Note that while the values of each run vary slightly, due to the random selection of the test and training set for each run, the results discussed below are typical of approximately 35 runs.

The model’s reported accuracy is 0.9078178927, or **90.8%**, which indicates a good model fit for the chosen data. However, it is a known behavior for decision trees of this type to suffer from potential overfitting, and we do not wholly discount that possibility here. We do not discount that this is partly influenced as the overwhelming occurrence of the *365,365* combination value of *age/duration*, which is the default value for the majority of the portals that make use of the CARE web services. As we have seen in earlier analysis, though this is a popular (and statistically most occurent) value, there are other significant combinations of these values that are also part of the dataset.

While the *F1* and the raw accuracy scores are often used to determine model fit accuracy, the *Matthews Correlation Coefficient* is preferred, mainly due to its consideration of the entire set of the confusion matrix values and penalization if the classification process does not do well on both positive and negative classifications [1]. The coefficient is calculated by the following formula:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (4.2)$$

As before, the exact value for the MCC differs for each simulation run due to the random selection of test and training values, but the aggregate value is approximately 70%. No instance of the roughly 35 test runs dipped below 69.5%. Given this value ($\sim 70\%$), the model does not appear as overfit as the raw accuracy score might allude, and beats a random choice for prediction of the outcome. While not an entirely definitive indicator of model efficacy, we can reasonably conclude that our model is a fair predictor of user behavior while not being hindered with poor false positive/false negative performance.

4.3.3 Recommendation Generation

After the creation of the decision tree, which includes hundreds of decision paths leading to leaves that result in an exported or non-exported classification, the base data are available with which to construct the recommendations using the session data as a basis. Each of the sessions is uniquely identifiable, which allows for a sequential recreation of all of the operations that took place for each session.

Essentially, the goal is to replay the session, simulating giving recommendations at each point that are based on paths that lead to a subset of the decision tree that terminates in a leaf that was exported to Excel, and to capture the efficacy of the recommendations given to the user. More specifically, the steps for creating and evaluating recommendations at each step in the session are:

1. The first operation is captured as the start state.
2. Starting with the second operation, the current state is compared to the previous state, creating a delta (Δ).
3. The Δ is translated into the feature extraction format represented at each node of the decision tree.
4. Using a separately created data structure that contains all of the node ID's with the fingerprint each distinct state, all of the decision nodes in the tree that correspond

with the current Δ are used as start points to navigate to leaves that are classified as exported (using a separate lookup table that identifies every node as either leading or not leading to an export. This generates a list of items to recommend to the user, representing user actions in ADVANCE.)

5. At each point, these recommendations are evaluated for the current Δ . To allow for a variable reduction of the number of recommendations (given that there are multiple paths, and the paths tend to be rather long) a filter was introduced as variable threshold value, which we call the *Recommendation Impact (RI)*. This value is calculated by taking the localized impact value of the node and dividing that by the total node impact of the entire tree. This process is notated in Equation 4.3. Given the choice of configured *RI*, the recommendation system will ignore any values that fall below this threshold.
6. When a column of the user actions in the current delta matches a given recommendation from the previous delta, the index placement in the session and the action are retained as a *taken recommendation*. We keep the full set of recommendations that were given to the user for the entire session, so any given recommendations that appear in the user's session *up to that point* are included as a *taken recommendation*.

To avoid circumstances where there are multiple values for *RI* for a particular recommendation, we also retained a current mapping of the *RI* to the unique combination of feature, comparator, and threshold. In instances where there are duplicate occurrences of the particular recommendation, we take the highest value of *RI* for the value that we've seen for this user's session and record that value as the *RI* for this sample. This is also intended to somewhat normalize the recommended input node to the path to an exported leaf by still recommending localized values for the current node. This does have the side effect of limiting the distribution of the values for *RI*.

In order to reduce the repetition of the same nodes being recommended repeatedly, TOUR Guide only recommends nodes from the current position to the leaf, not the path from the root to the leaf that includes this node. Initially, the entire path to the exported leaf was recommended to the user. This seemed appropriate given the obvious improved *RI* that would result by including all of the nodes from the root, but the practical result was that there was significant repetition in the recommendations given, including all recommendations having the root node in their recommended set. For this reason, and to reduce the candidate set of recommendations, we chose to omit any values higher than the current node’s place in the decision tree in the set of possible recommendations.

$$\textit{Recommendation Impact (RI)} = \frac{\textit{Local Applicable Samples}}{\textit{Total Operations}} \quad (4.3)$$

4.4 Recommendation Simulation Results

For our simulation, we used the configured extraction of 25% of the total sample data, taken for each of the unique recommendation sessions by randomly shuffling all of the session indices using Python’s random function, which uses a form of the *Mersenne Twister* PRNG [12]. Removing the sample operations, we continued to decrement the count of necessary operations to satisfy the 25% sample requirement and continue to extract sessions until this threshold was reached. While the amount of sessions differs for each run due to the random selection, we observed typical extraction being between 8,300 and 8,600 sessions.

Following the extraction of this simulation set, we further subdivided the remaining values (from the original set) into the test and training sets for the construction of the actual decision tree. For our tests, we used a setting of a 75% training and 25% test population split, which is a common split for decision tree analysis.

Although the exact number differs for each run of the simulation, for the simulation that we will be examining there were 70,741 total recommendations given by the system

that appeared within the user’s sessions. These recommendations were generated using the constructed binary decision tree which contained a total of 29,705 nodes.

To determine the result of the given recommendation, we used a look-back buffer to record the results of the previous level’s result in the tree. This yields a process that follows the pattern shown in Figure 4.7, wherein the tree is constructed in a way where decisions that split *left* are *true* and *right* are *false*. The decisions are organized by *feature*, a relational operator (which is always defined as \leq), and the *threshold*. For the purposes of evaluating the result at each point, we define a meta-node at the $n+1$ position (where n is the node’s tree depth) in the tree hierarchy that represents the decision result. The actual node is the next decision in the path to the leaf. This conceptualization allows for a parallel state machine to be kept to represent the current decision path that led to the current point which would otherwise be extracted by simply following the binary decisions of the tree from root to leaf.

$$OTR = \frac{1}{\text{Available Recommendations}} \tag{4.4}$$

Feature Evaluation Process

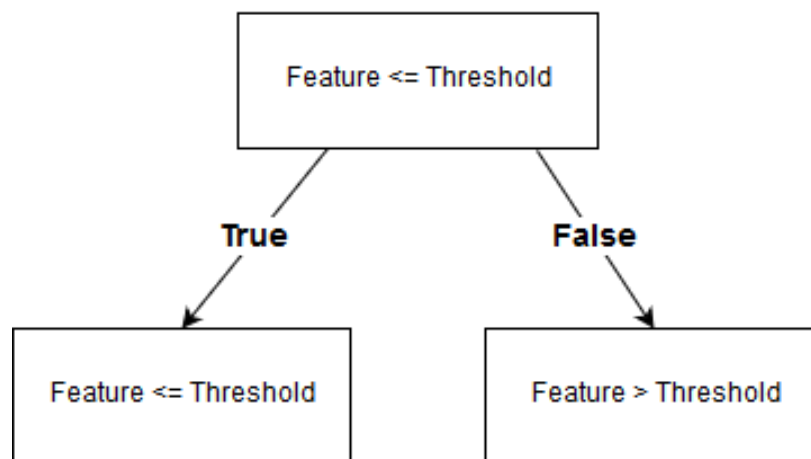


Figure 4.7: Summary of the Feature Evaluation Process. Each decision node is evaluated (shown in the top node), with the result relational operator (\leq or $>$) modified based on the decision outcome.

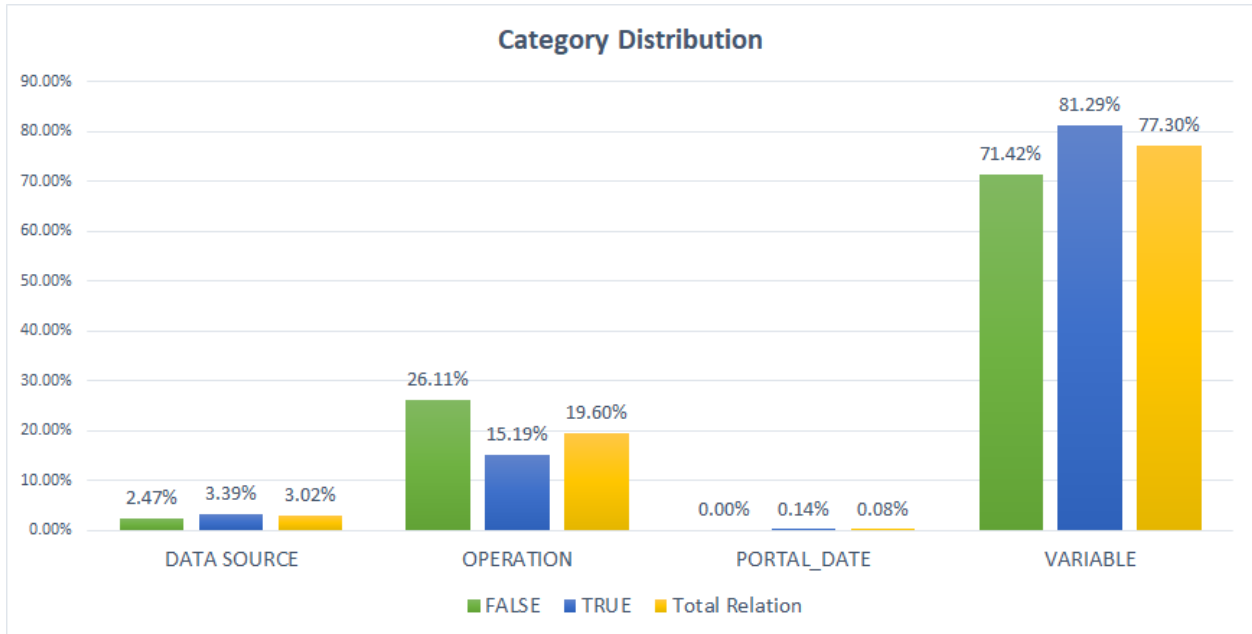


Figure 4.8: Distribution of the decision tree recommendations by category. TRUE and FALSE indicate whether the $n-1$ node's decision in the tree was taken. The *Total Relation* is the summary distribution of the categories, irrespective of decision.

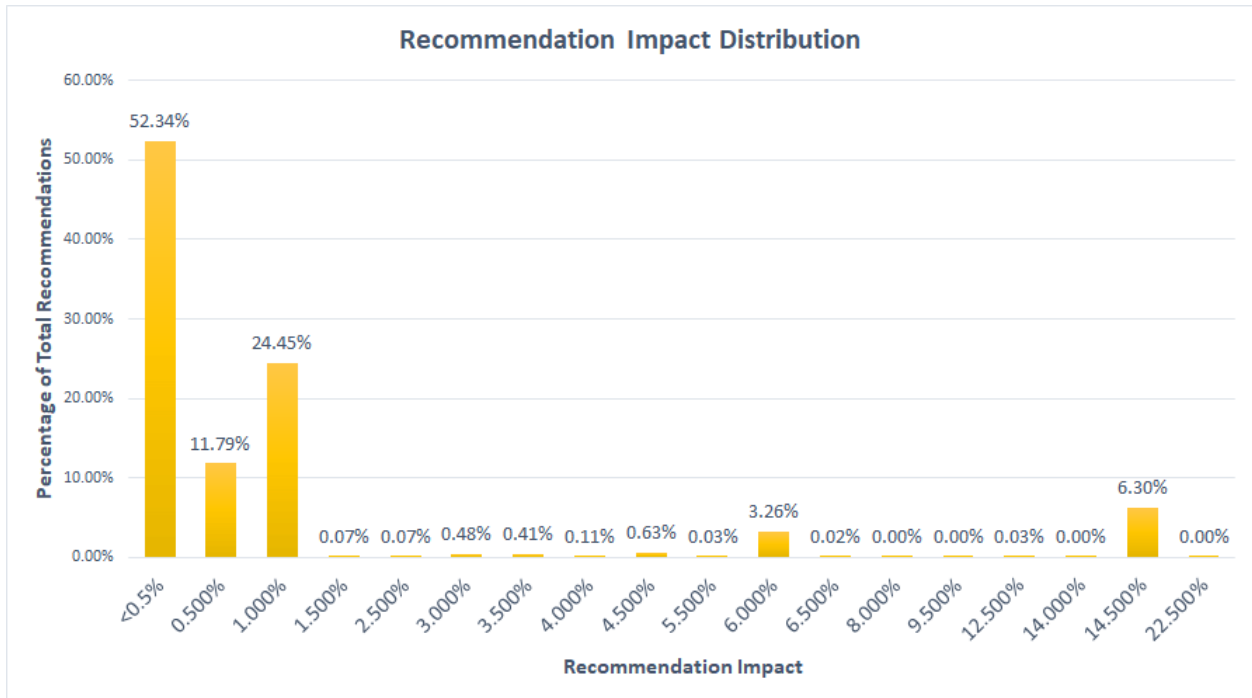


Figure 4.9: Distribution of the Recommendation Impact (RI) frequency in the total recommendation set.

Due to the various specific values for the decision node points that serve as the recommendations, there are thousands of individual values that were part of the recommendations. To distill the analysis to the *types* of recommendations that we see instead of focusing on the specific values, we created and then divided the recommendations into the following four primary categories:

1. Data Source
2. Operation
3. Portal Date
4. Variable

These categories directly reflect the sources that combined to yield a single operation row representation. The Data Source category represents the currently selected data source that provides the context for the overall analysis. The Operation category records the distinct values for the possible operations within the system, most of which are related to set manipulation to filter and change display of variables. The Portal Date category captures the values of weekday, hour of day, age, and duration. As these are the only values that are considered on a non-binary categorization, and deal with time (either as part of the analysis directly or as metadata), they are considered together. Finally, the Variable category represents changes to the individual variables either displayed or manipulated through the processes available in the portal. These values are made up of all of the finite-valued (i.e. non-free-text) values within each of the datasets. These values represent the bulk of the individual values that are part of the tree node structure.

Because the array representation allows for more than one facet (represented by the column in the row) of the operation to change from the previous operation, it is possible that there is more than one value that would trigger a recommendation for each operation delta during the simulation process. For this reason, there may be overlap in the recommendations given at each point. However, the logging system for the recommendations

captures each of the individual recommendations as an individual item while also recording the ordinality of the recommendation within the session. This will allow us to identify any coincident recommendations given as the result of multiple items within the operation's Δ .

The breakdown of the distribution of the categories is shown in Figure 4.8, which includes the distribution of values that were reached as the result of a true path (decision was followed), false path (decision was “rejected”), and the sum of either path (ignoring how the category value was reached).

Overall, we see only small deviations in the differences between these three distinctions internally. However, interestingly, we do see some differences from the distribution of the total proportion of the values when compared to the raw distribution of the total features. For example, within the Variable category we observe a raw membership of 88.2% (718 of 814 total features), an 11% raw change from the category presence in the actual recommendations. A successively larger set of changes are noted among Portal (36%), Data Sources (-172% deviation), , and finally Operation (-921%). These discrepancies highlight the relative impact of the categories within their recommendation frequency and how they differ from the simple numerical weight of the feature's volume in the set of all features.

Looking at the distribution of the *RI* for the recommended values (shown in Figure 4.9), there is an overwhelming majority of the recommended values that have values for *RI* less than 1.5%, with 88.58% being in this category. As stated previously, we intentionally modified the original algorithm from originally recommending all items in the path (all the way from the root to the leaf) to only provide recommended nodes from the current node to the leaf of the tree for the multiple paths that would take us there. As a consequence of this change, the values at the top of the tree are less likely to be chosen (unless the user picks those exact values), which has the impact of lowering the likelihood of large values for *RI* being present in the set of recommendations. The variations that we do see that in-

clude these values likely stems from the user choices that place the recommendation within the specific tree location.

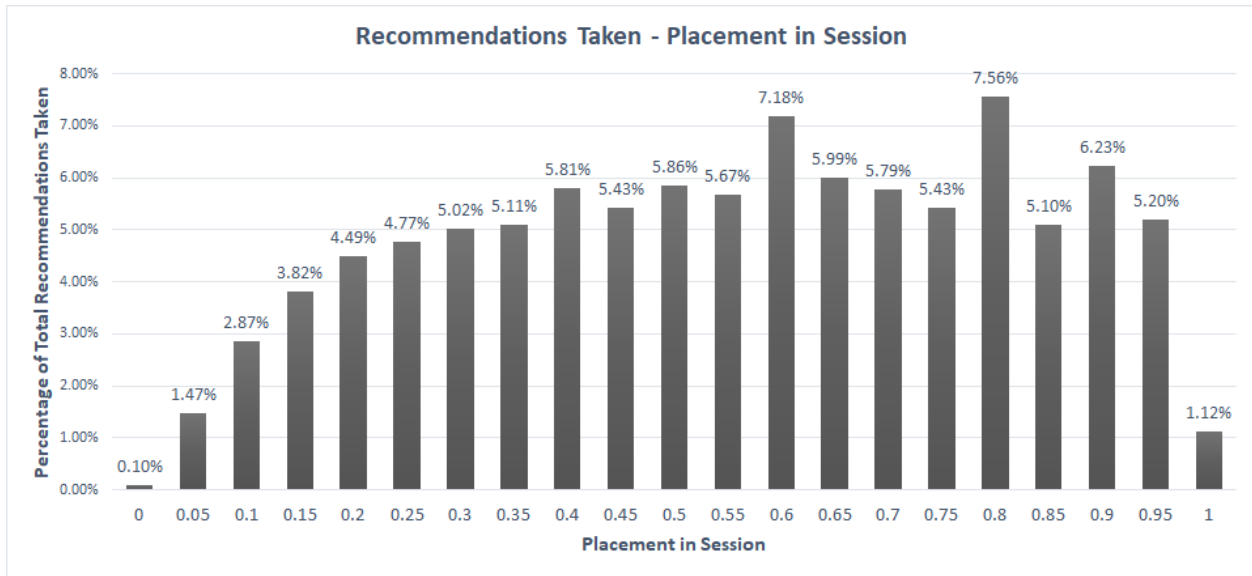


Figure 4.10: Distribution of the recommendations taken, by placement in the overall session.

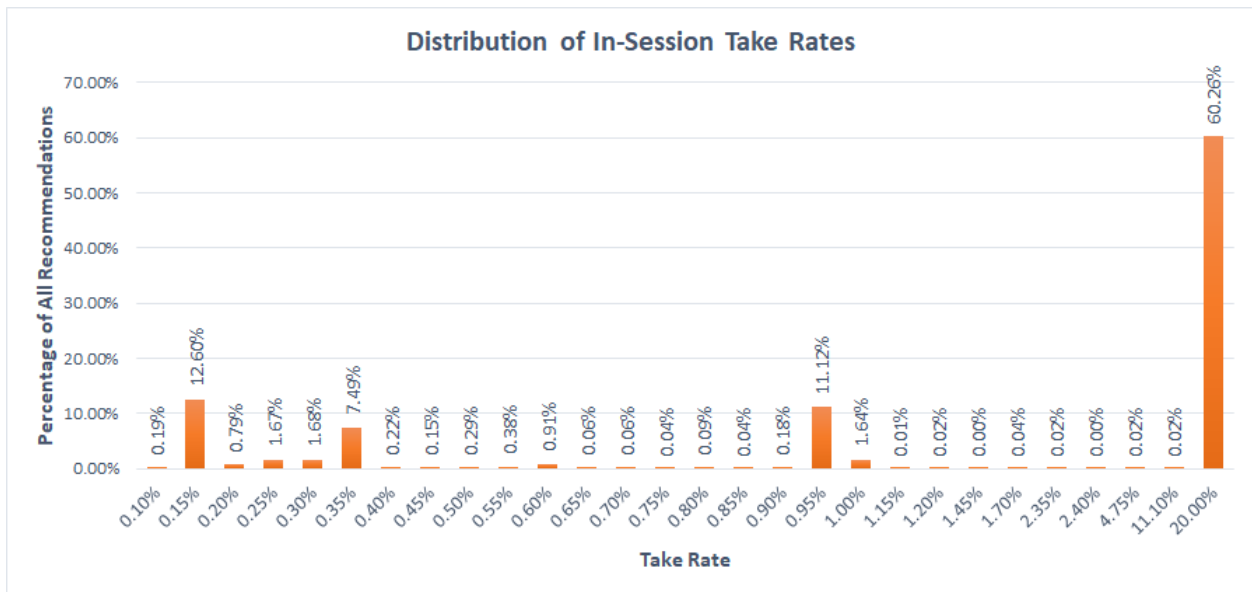


Figure 4.11: Distribution of the raw in-session take rates.

$$x = RI \times OTR \tag{4.5}$$

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)} \tag{4.6}$$

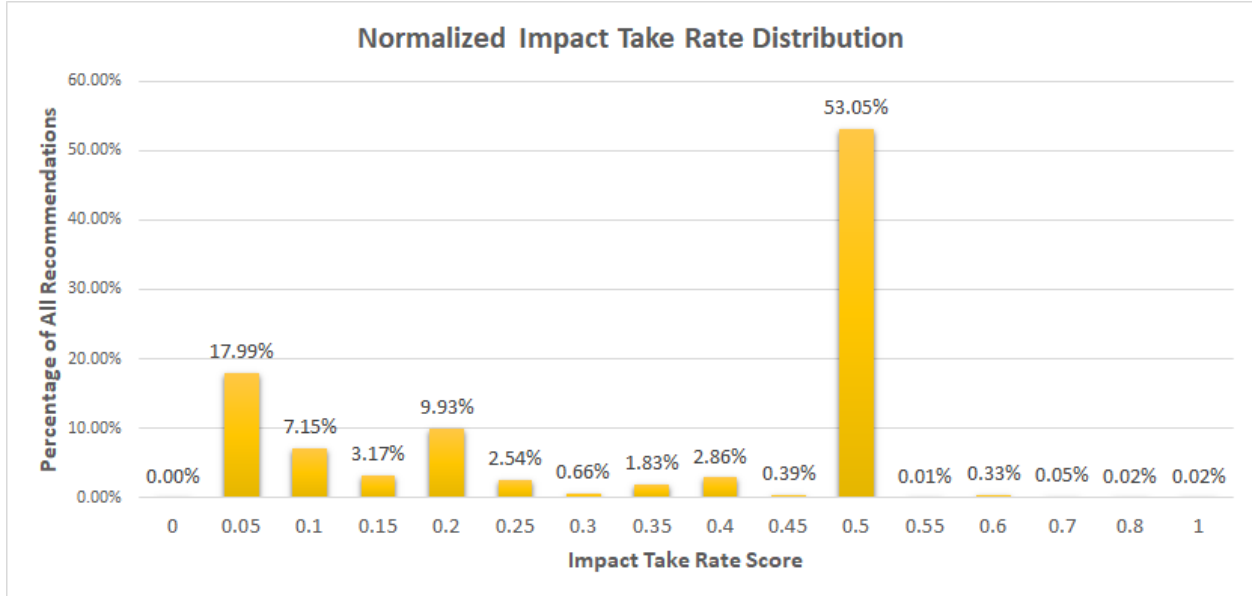


Figure 4.12: Distribution of the normalized in-session impact-adjusted take rates.

At each point in the recommendation simulation process, we captured the index of each of the recommendations that were taken. Using the total count of operations within the session, we can determine where the recommendation was taken within the user’s session. We see from the results (shown in Figure 4.10) a sharp increase in the taken recommendations from the beginning of the session, with a flattening of the trend at a little before the 40% session mark. Note that these session markers are inclusive at each point, meaning that the marked ranges go from the rounding point at each bin (e.g. actual values included that are included in the bin for 0.1 range from 0.05+ to 0.124...). The values are binned in this manner to make display of the values easier, given that the distribution of the session occurrences are broad in their specific values. This behavior diverges from previously observed high engagement (measured with relatively high time spent on page (TSP) for that user’s session) behavior that showed significant spikes in user engagement. This basic examination of the recommendation taken location seems to indicate no major

periods where the recommender system was providing recommendations that were equally useful across the entire session, at least as measured across the behavior of the entire population.

To measure the presence of recommended values within the user’s total session, we define the metric of *operation take rate* (OTR) which represents the single-operation rate for that given recommendation scenario. This is defined in Equation 4.4. Within these values, only seven of the total binned values rise above 1%, with only four of those exceeding 2% (with take rate in parentheses): 0.15% (12.6%), 0.35% (7.49%), 0.95% (11.1%), and the largest value of 20% (60.26%). With such a large single occurrence percentage, we wanted to more closely examine the recommendation characteristics of scenarios with this take rate, so we extracted all of the recommendation records with this value to determine if there were any explanatory values within the collected fields.

Inside the group of recommendation rows with this 20% take rate, because of the definition of OTR (as defined in Equation 4.6), all of these scenarios had five available recommendation options when the recommendation was seen within the user’s next operation as being chosen. Looking at the set of values in this set, all of the recommendations are within the *variable* category. In fact, all of the values that make up this group are the same value. This value is specifically a variable that captures the date of issue of a traffic citation, and for all of the recorded taken instances of this recommendation, the value was recommended to *not* be chosen as part of the path. Although the value for the take rate is the highest observed, the corresponding RI value is 0.0072%, one of the smallest observed values for this score. Considering the inverse relationship of the RI and OTR, we determined that a more holistic value for determining the overall impact that the recommendation had on the system was necessary.

Based on the above observation (and some other less extreme examples within the data), we calculated this *impact-aware take rate* to provide a weighted measure of the take rate that considered the amount of elements that the individual recommendation had the

potential (by its value for RI) to affect. This modified rate is defined as the product of the recommendation impact and the operation take rate for each operation (shown in Equation 4.5). Given the small values for each of the individual operands the resulting values are smaller still. Because of this, we also re-scaled this value to fall between 0 to 1 using a typical min-max normalization (as shown in Equation 4.6).

The values for the re-scaled (normalized) impact take rate are shown in Figure 4.12. To begin, no significant amount of the re-scaled values rise above 0.5 within the group of all of the recommendations given, with the total of all values above 0.5 making up less than 1% of all recommendations. Also, while not a directly comparable set due to the observed variability of the of the RI and OTR values, looking at both Figure 4.11 and Figure 4.12, the influence of the prevalence of the 20% take rate in the raw value is clearly seen in the coincident spike within the adjusted take rate values. Breaking down the values for this adjusted take rate, we can see only a few other notable spikes (with percentage occurrence in parentheses): 0.05 (17.99%), 0.1 (7.15%), 0.2 (9.93%), and the aforementioned 0.5 (53.05%).

4.5 Conclusions and Future Work

4.5.1 Conclusions

With TOUR Guide, our goal was to provide a locally-aware recommender system to optimize for decision paths that led to exporting the results of an analytics session to Excel or CSV. This system is intended to be configurable and expandable to not only the ADVANCE system, but other systems that center around distinct and purposeful analytics sessions. Through examination of a simulated recommendation run of a subset of the sessions within the system, we can extract some general findings based on the observed interactions between the provided recommendations and the contents of the users' actual sessions.

We feel confident that we can put forward that data export (in this case to Excel/CSV) is a reasonable preference indicator and can serve as a reasonable optimization basis for a recommendation algorithm. This conclusion stems from the observation that users show a trend towards export as their session position increases. This seems to indicate that this export process, at least over the general population, represents a concluding event within the session and indicates that users work toward that outcome as a goal in the observed sessions.

Based on our observations of both RI and OTR, we recognize that the relationship between raw impact and the rate at which a recommendation was taken (i.e. the popularity of that recommendation) are not sufficient explanatory tools when considering the efficacy of this kind of recommendation process. Ideally, we would have liked to have seen large values for RI correspond with the most popular items, indicating an early influencing of the user's session toward a path that would end in an export. However, what we saw instead was a complicated and often inverse relationship between these two values. This indicates that, for the sessions we observed, the values that were put forward as recommendations were often those very far down within the tree itself and thus had very low values for RI. This was obviously influenced by our decision to ignore items above the current item's hierarchy, which had the effect of ignoring upper-level tree items to avoid giving the same recommendations over and over. There may be room to consider a hybrid approach that recommends some of these higher-level elements, but also possibly taking into account the path similarity within the user's session to only recommend items that were present in one or more of these paths from the root. This pattern of overall low RI values also highlights the impact that the user activity extraction and classification process can have on both detection of and building recommendations for the user. Specifically, the very exact nature of this process appears to add a level of 'stiffness' to the recommendations, especially considering that the categorical or semantic similarity of variables were not a part of the selection process. This would need to be considered in the expansion of this process

to improve the relevancy and intuitive help that this system is intended to provide to the user.

Looking further into this relationship using the impact-aware take rate, we see that the overall behavior of the taken recommendations in the system are in the bottom half of the ‘popular and impactful’ set of possible taken recommendations. Although having consistently high values for this intersecting metric isn’t a fool-proof indicator of good recommendation performance, it does lend evidence to the conclusion that users are not acting in a way that the strict paths would send them down if they had followed the recommendations exactly. Users, especially those that can have preferences that are role-based (within their organization) and that can change mid-session, are difficult to classify within the narrow parameters defined by the path-based recommendations. This doesn’t mean that there isn’t value in these recommendations, especially since the system is responsive to each change that the user makes within their session, but that more context-aware recommendations (based on some level of intent-based explicit feedback from the user) might improve the performance of a system that relies heavily on specific user decisions as input and then provides recommendations at the same specificity as output.

4.5.2 Future Work

Expanding Preference Candidates While we did see that export of values in the system as appearing to indicate preference, by assuming that users cease using the system when they have reached their goal (or simply given up in the pursuit of it), we would like to further study the effects of additional preference indicators. Among these that are available in the system is a more advanced interface for creating very detailed slices of the data. This interface, called a ‘drill down’, allows users to proceed along a path that sees the set of values reduced by the application of successive filters. We consider it strongly possible that engaging in this process, even absent any other metrics for preference (e.g. linger time or export) implies a focused interest in a particular set of filters and variables.

We did not examine this process due to technical limitations of the logging process, but with adjustments to that process, this procedure could be studied and would possibly yield interesting data regarding similarities and differences in usage patterns for simple exports when compared to the drill down process. We would also like to explore the inclusion of conceptually similar items within the category of the current node being recommended as well as using context for the session (e.g. what deployed site the user is visiting the system from) to promote a context-aware approach[8].

User Trial and Settings Modification Due to the passive and non-interactive setup of this research, a directly user-involved trial of this method for directing users to items that have an impact in leading to an export of the data is a logical next step for this process. Given the complexities and potential impact of interfering with the day to day business of these users, who are acting as part of a role instead of simply expressing personal preference, we consider this a process that should be approached with extreme care. However, we do consider this as a significant source of information to not only elucidate more accurate behavior results for our method, but also to aid in providing feedback for the interface for providing these recommendations. The values for RI could be used to potentially limit the presented values to the user, which would allow for modification of the inclusion criteria to add more values. The TOUR Guide system already has a configurable value for a minimum threshold for the RI, so manipulation of the selection criteria would be easily filtered with only a modification to this value, allowing for fast iterative experimentation of the impact of these lock-step changes. As part of improving the construction of the tree, we would like to also consider investigating different methods of entropy calculations, as there has been some work related to less harsh Gini calculations that have seen promising results[5] and may help avoid artificial decision breaks within the tree.

References

- [1] BALDI, P., BRUNAK, S., CHAUVIN, Y., ANDERSEN, C. A., AND NIELSEN, H. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics (Oxford, England)* (2000).
- [2] BENNETT, J., AND LANNING, S. The netflix prize. *Proceedings of KDD cup and workshop* (2007), 3–6.
- [3] BOUZA, A., REIF, G., BERNSTEIN, A., AND GALL, H. SemTree: Ontology-based decision tree algorithm for recommender systems. In *CEUR Workshop Proceedings* (2008).
- [4] BREIMAN, L., FRIEDMAN, J. H., OLSHEN, R. A., AND STONE, C. J. *Classification and Regression Trees*. 1984.
- [5] CHANDRA, B., AND PAUL VARGHESE, P. Fuzzifying Gini Index based decision trees. *Expert Systems with Applications* 36, 4 (2009), 8549–8559.
- [6] GERMAIN, A., AND CHAKARESKI, J. Spotify Me: Facebook-assisted automatic playlist generation. *2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)* (sep 2013), 025–028.
- [7] GOLBANDI, N., KOREN, Y., AND LEMPEL, R. Adaptive Bootstrapping of Recommender Systems Using Decision Trees. *Wsd 2011* (2011).
- [8] HARIRI, N., MOBASHER, B., AND BURKE, R. Context adaptation in interactive recommender systems. In *Proceedings of the 8th ACM Conference on Recommender systems - RecSys '14* (New York, New York, USA, 2014), ACM Press, pp. 41–48.
- [9] KONSTAN, J., RIEDL, J., BORCHERS, A., AND HERLOCKER, J. Recommender systems: A groupLens perspective. *Recommender Systems: Papers from the 1998 Workshop (AAAI Technical Report WS-98-08)* (1998), 60–64.
- [10] KONSTAN, J. A., AND ADOMAVICIUS, G. Toward Identification and Adoption of Best Practices in Algorithmic Recommender Systems Research. In *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation* (New York, NY, USA, 2013), RepSys '13, ACM, pp. 23–28.
- [11] LORENZI, F., LOH, S., AND ABEL, M. PersonalTour: A recommender system for travel packages. *Proceedings - 2011 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2011 2* (2011), 333–336.
- [12] MATSUMOTO, M. Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator Dedicated to the Memory of Nobuo Yoneda. *ACM Transactions on Modeling and Computer Simulation* 8, 1 (1998), 3–30.
- [13] PARRISH, A., DIXON, B., CORDES, D., VRBSKY, S., AND BROWN, D. CARE: an automobile crash data analysis tool. *Computer* 36, 6 (jun 2003), 22–30.

- [14] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., AND DUCHESNAY, É. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* (2012).
- [15] PORTUGAL, I., ALENCAR, P., AND COWAN, D. The use of machine learning algorithms in recommender systems: A systematic review, 2018.
- [16] RAILEANU, L. E., AND STOFFEL, K. Theoretical comparison between the Gini Index and Information Gain criteria. *Annals of Mathematics and Artificial Intelligence* (2004).
- [17] SMITH, R. K., GRAETTINGER, A. J., KEITH, K., AND PARRISH, A. Identifying High Frequency Crash Locations: Empowering End-Users with GIS Capabilities. *ITE Journal* 77, 1 (2007), 22–27.
- [18] WANG, H., PARRISH, A., SMITH, R. K., AND VRBSKY, S. Improved variable and value ranking techniques for mining categorical traffic accident data. *Expert Systems with Applications* 29, 4 (nov 2005), 795–806.
- [19] YOON HO CHOA SOUNG HIE KIM, J. K. K. A personalized recommender system based on web usage mining and decision tree induction. *Expert Systems with Applications* (2002).
- [20] ZHANG, H.-R., AND MIN, F. Three-way recommender systems based on random forests. *Knowledge-Based Systems* (2016).
- [21] ZHEN, L., HUANG, G. Q., AND JIANG, Z. Recommender system based on workflow. *Decision Support Systems* (2009).

CHAPTER 5

CONCLUSION

Going forward, the reliance on the use of data-driven decision-making tools is one that is not likely to see a reduction in either importance or frequency. Data's use as a tool for informed decision-making continues to expand to nearly every industry and domain, including some that either have not necessarily been historically considered as having relevant data to analyze or those that did not see data as a particularly valuable asset in their day-to-day. As more and more data are collected, systems that exist to allow the extraction of that data, and especially systems that support synthesis of that data into valuable *information* (which is different than simple data) become vital for those in these organizations that are tasked with making sense of their data, whether it is collected as a primary function of their operation, or simply as associated metadata for their activities. Through the analysis of user behavior, construction and examination of education and recommendation systems, and consideration of the impacts of the confluence of all these efforts we can see certain patterns and emergent conclusions that are relevant to the design of these systems and the ancillary systems that support them. Through our work to **understand**, **educate**, and **optimize**, we were able to discover characteristics of these systems and how we can augment this process through the insights provided by that comprehension.

5.1 Understanding Overall User Behavior

Beginning with the analysis of users' behavior, we were able to determine that for the majority of users, their interactions with the system are relatively brief and purposeful.

Users visit the system, extract their desired information, and (for the most part) do not linger for extended periods that aren't in support of that process. This behavior highlights the importance of not only the choice of defaults within the system (which seed the user's behavior throughout their session), but several other factors as well. First, the efficiency of the system, especially as it relates to information retrieval performance, is extremely important. For users that on average spend **3.6 minutes** in a session, any delay in providing data extends what should be a laser-focused session into a frustrating and potentially damaging blow to the user's trust of the system as a reliable tool. It is not uncommon, especially for reports and data extraction that must be based on the latest possible data, for investigation sessions to take place extremely close to the due date for such information. If the system repeatedly slows down, or otherwise inhibits use, the system is possibly impacted in a way that could lessen the user's trust in and reliance on data-driven decision-making tools.

In addition to the typical user that reaches a goal quickly, we do observe a non-trivial portion of the users that spend quite a long time within the system (defined as 26 or more operations in a session). As we saw during the examination of sessions of differing lengths, users on this longer timeline tend to have a linearly increasing trend toward higher engagement in their session the closer they get to the end of their session, regardless of the overall length. In fact, across both short and long sessions, there is a tendency to have at least some increase of high engagement periods toward the end of the session. In short, we see that users, regardless of their specific session length, have repeated patterns of low engagement followed by periods of relative high engagement. This indicates that users are engaging in purposeful examination, characterized by low engagement setup periods (clicking around the interface to focus on the data they want to see) followed by longer, higher engagement periods wherein they examine the fruits of their labor.

From this we can extract several possible ideas for improving system function, using the increase of the frequency of periods of high engagement as a goal. Among these are:

1. Designing the interface in a way that reduces the number of user interactions (clicks) to get to a minimally viable result (i.e. a filtered dataset).
2. Carefully considering default value selection, with a specific effort to reduce possible bias.
3. Implementing a set of ‘analytics packs’ which use the high engagement periods as identifiers for item preference characteristics.

As part of designing the interface for result-to-interaction efficiency, considering the defaults that are necessary to accomplish this goal is critical. While we did not see extreme evidence of non-deviation from the age and duration default values (i.e. users did choose other things than the defaults), we would caution designers of these types of systems to carefully consider their choice of defaults. These defaults should be based, as much as is possible, on empirical data of user preference and not just anecdotal scenarios shared by a subset of users. The introduction of potential bias into the outcome of the user’s behavior, especially in decision support and analytics systems, can have impacts not seen in simple preference-based systems that seek to optimize toward a user’s personal desires instead of their practical needs. Complementary to avoiding general bias through defaults is how to design a system in such a way that can use the user’s behavior to promote knowledge and education, which we saw through our education and Markov chain based recommendation process.

5.2 Education and Awareness

When entering a new place, taking a quick drive around is a common way to acquire a general ‘lay of the land’ that is then filled in with more detail over time. This extended proprioception, while sometimes difficult to explain, contributes to a generally increased sense of place and location awareness. As with any new environment, people require time to get acclimated to the particular characteristics, rules, and idiosyncrasies that exist within

it. This concept is transferable to that of exploration of datasets using data mining and analytics systems. Regardless of the effectiveness of the tool, having the user have a reasonable mental map of the general outline of the data, as well as an understanding on how to navigate it, gives a baseline for critical thought and idea linking that can come from familiarity-grounded ideation.

With this in mind, providing users a data-driven means by which to explore the popular and less-visited areas in a data system can provide a much less meandering and random approach to data familiarity. We found, through the use of Populist and Novel methods, we can build an always-localized set of answers to the question “Where should I go next?” without simply recommending the same popular places (i.e. items) or, conversely, simply pointing the user to the most abandoned place within the data.

Our analysis of the behavior and subsequent benefit of these Populist and Novel methods for both popular and lesser-visited items indicates that these are promising methods for organizational education. While we recognize that nudging users toward the extreme ends of popular and unpopular items can be problematic, we would qualify that the overall target for a Populist method would be for a user that needs to learn (or relearn) the data environment quickly. Alternately, the Novel method would be targeted to more seasoned user who could benefit from introduction to items within the data that they may not have visited, but that may still have value, even if they determine that they weren’t visiting the items for a reason. An active rejection of an available choice is one that is based on reasoning, rather than ignorance.

5.3 Preference

User indications of preference are important tools in constructing models that can provide relevant recommendations to a user. However, the input data must be an accurate indication of the user’s current preferences to be truly effective. Detection of this preference, especially in cases where a user changes their intended goal or overall pattern of use,

is extremely difficult since there often isn't a clear and immediate inflection point that signals this change. In cases where it is available, implicit indicators of preference are less than ideal. Within the ADVANCE system, and in many other data exploration systems, recommendation is a secondary goal that leaves less room for explicit preference collection. These can take several forms, but are sometimes implemented with either binary indicators (good or bad) or give the user a scale on which to rate the data they are being shown.

Without an explicit 'thumbs-up' or 'thumbs-down' indicator for a given data result, we have to rely instead on a few preference indicators that we can extract from the user's behavior. The two methods that we see, user linger time on the page and the user's choice to export the data, both provide their own benefits and caveats.

Raw linger time, especially given the broad range of observed behaviors by the users, is a relatively poor (or at least inaccurate) indicator of preference for many reasons. Examining the relative linger time within the session (i.e. high-engagement) can give us an idea of what the user prefers relative to other items within the session, but there are a number of reasons that can lead to high relative linger times, many of which are not indications of preference at all. Among these are included the numerous 'away from keyboard' variations, even those that still see the user still at the computer. Especially with the proliferation of workstations with more than one monitor, the possibility that the user has switched to another application on another monitor increases.

Recognizing linger time for the limited preference indicator that it is, we place much more confidence in the explicit user export of data, whether that be to a screenshot, image, or to Excel or CSV. This is an activity that the user must trigger manually, and the export format implies that the data is interesting enough to warrant further examination. Although, in all cases, we are reliant on implicit feedback from users that doesn't seek any input from the user. This has obvious benefits in the volume of data that it allows to be collected, but ultimately removes the user from the engagement process as an active participant in the system. This is important to remain aware of when constructing systems

that recommend items or paths to users, especially when considering the real-world potential consequences of their decisions based on this data.

5.4 Recommendations and Optimization

Within the recommender system domain, one of the core concerns is to determine and optimize the system toward individual user preferences. However, for data analysis systems like ADVANCE, the use patterns are somewhat different, especially since these are work-related tools that are used professionally that use multiple types of public safety data. Considering these differences, we see the following characteristics related to behavior and recommendations:

1. Users in these systems often (though not always) act in a way that is in support of an organizational goal, not necessarily their own preferences.
2. This role-centered behavior lessens the effectiveness of recommendation processes that prioritize user preference.
3. Methods that educate users and promote role-supporting workflows have shown some promise in making user interactions with analytics systems more efficient.

As alluded to in the earlier discussion on education related recommendations, the promise of providing an education tool that could both “quick-start” new users and provide a “deep-dive” for long-time users can be fulfilled by providing item recommendations that are curated for each goal based on the weighted pattern behavior of users of the system. In this case, those recommendations were provided by using a Markov chain representation of all of the activity of the system, which can be subdivided by creating purpose-built chains (based on different website frontends, for instance) to provide recommendations that are based on the behavior of that subdivision criteria. Based on the system that we constructed based on this concept, we found that although the overall passively observed behavior did

not align directly with the recommendations provided by the Populist method (which provided recommendations from popular items based on what the user was currently viewing), this stemmed from the individual variability that was present within user sessions. Despite this specific situation, we see much potential for use of this type of system for orientation-style navigation suggestions. Among the Novel recommendation method, we did see an expectedly low uptake of the values provided by this particular process. Even so, because we were limiting both already seen and already recommended values, we concluded that removing this already-recommended restriction for this method would likely not negatively impact the user in the way that it would for the Populist method. In addition, since the goal of the Novel process is to improve the overall low score visited nodes, repeated recommendations (as long as we don't recommend things that the user has already done) would promote this goal.

In order to solve the workflow optimization problem with these types of systems, and with the accuracy problems that arise related to implicit feedback like time spent on page (TSP), we created the tree-optimized user recommendation (TOUR) Guide system to use the decisions related to the binary classification of exporting values from this system as input to a recommender system. After a full simulation of creation of recommendations using real user sessions, we observed most of the paths that users took were those that were much closer to the terminal leaf of the decision tree than we might have anticipated and that a majority of the paths that were popular (were 'taken' by the users within their session) were also found in nodes that had low impact scores. Overall, we determined that while this method overall has promise in providing relevant, workflow-optimized recommendations, it would benefit from an adjustment to the initial restriction that we had imposed on 'from this point downward' nodes, where only nodes below the current node in the tree were recommended to the user. This was done to avoid repeatedly recommending the same items to users, and did limit the amount of duplicate recommendations, but also

limited the recommendation impact (RI) of the nodes and resulted in a very focused set of recommendations overall.

5.5 Considerations and Summary

Due to the nature of the type of data that has been collected, several potential issues exist which will either need to be considered for (or at least recognized during analysis) to help avoid drawing incorrect conclusions. The implicit collection of data makes elucidation of true intent difficult, especially given that anecdotal evidence seems to suggest that similar activities can be performed by a user with very different intents (e.g. meandering aimlessly through the data just to be familiar looks often like an attempt to answer a specific question, but really the user is just exploring the system’s data and capabilities). Collection of explicit feedback is possible, but (as with other systems aiming to recommend items), care must be taken to adjust to the user’s willingness to contribute to the system attempting to cater the results to them. Being too overbearing in the process can be annoying to the user and act as an obstacle to building and retaining trust in the recommendation system. Conversely, the system must be constructed to trust the “right” users (ones that provide good feedback) [4]. Decision trees have a tendency to overfit data during the training process[1], making accurate predictions (that have global impact) difficult. The use of other accuracy measures (e.g. Matthews Correlation Coefficient) seeks to identify significant biases in systems of this type, but cannot totally eliminate the problem.

Even among users that may be more similar in goal and behavior due to their use of the system in an organizational role, different users will likely prefer different levels of assistance. It is feasible that a subset of the users of the system would prefer to opt-out of the assistive features of the system for a number of reasons. If disabling of the assistive system is added as a feature, it might follow that the collection of the user’s data would be appropriate and reasonable. However, the information about behavior that would have been passively gathered would not be available to the system, lowering the overall accu-

racy of the model by limiting the available data to only those that choose to allow this collection and further opt to benefit from it.

Further, allowing for partial opting-in is problematic. If the system benefits from more data about the user, then less data (or even no data) is likely to lead to a less helpful system. This side-effect would reinforce a user's belief that the system was ineffective and potentially lower overall user-base trust in its abilities. It also has the problem of being cyclical and becomes worse the more it occurs. The user opts out of the system, then opts back in only to find a system that is ill informed as to the user's preferences, which continues until the user either gives up or allows for enough failures to build up the necessary amount of data to become helpful.

Public safety analytics and decision support systems present unique opportunities for research, in particular due to the relatively low public research volume of these systems and datasets within the domain of recommender systems, the potential for knowledge-gain and collective improvement is encouraging. The continued study of these systems, especially with a goal of simultaneous contributions to data-backed suggestions for both their direct design and toward that of systems that support them, demonstrates a need for focused analysis of available public and private systems in this category. The humble hope for this research is to serve as a potential example for mixed-outcome study that can be applied across both research and practical domains for systems of this species, and to improve both the function of these systems and the decisions that emerge from their use.

REFERENCES

- [1] BRAMER, MAX, *Principles of Data Mining*, 2007.
- [2] BREIMAN, L. AND FRIEDMAN, J.H., OLSHEN R.A. AND STONE, C.J., Classification and Regression Trees, *The Wadsworth Statistics Probability Series*, 1984.
- [3] KANTOR, P., ROKACH, L., RICCI, F., & SHAPIRA, B. (2011)., *Recommender systems handbook*.
- [4] O'DONOVAN, JOHN AND SMYTH, BARRY, Trust in Recommender Systems, *Proceedings of the 10th international conference on Intelligent user interfaces - IUI '05*, 2005.
- [5] PARRISH, LS AND DIXON, BRANDON AND CORDES, DAVID AND VRBSKY, SUSAN AND BROWN, DAVID, CARE: An automobile crash data analysis tool, *IEEE Computer*, Volume 36, Number 6, pp. 22–30, 2003.
- [6] SMITH, RANDY K. AND GRAETTINGER, ANDREW J. AND KEITH, KERRI AND PARRISH, ALLEN, Identifying High Frequency Crash Locations: Empowering End-Users with GIS Capabilities, *ITE Journal*, Number 1, pp. 22-27, Volume 77, 2007.
- [7] STEIL, D. A., PATE, J. R., KRAFT, N. A., SMITH, R. K., DIXON, B., DING, L., & PARRISH, A., Patrol routing expression, execution, evaluation, and engagement., *IEEE Transactions on Intelligent Transportation Systems*, 12(1), 58-72.